11

Smart Zones to Create the Ambience of Life

Etienne de Sevin, Caroline Chopinaud, and Clodéric Mars

11.1 Introduction

To design "background nonplayer characters" that breathe ambient life into a virtual environment, we propose new concepts that facilitate the creation of *Living Scenes*. The aim is to generalize their use in video games as a way to improve the feeling of presence for the player. This chapter introduces the concept of *Smart Zones* used to design credible, consistent, and interactive ambient life, involving autonomous and adaptive NPCs.

11.2 Designing an Ambience of Life

Consider the following situation: *8 pm, the night is still young when our player arrives at a restaurant looking for his friends. As he enters the building, he sees dozens of clients eating, talking, and drinking. His friends, as they see him, made signs for him to join them. On the way, he meets a waiter asking him if he wants to order something.*

The problem we are trying to solve is how to quickly structure and design this ambience of life. Our answer: using a set of Living Scenes!

11.2.1 What Is a Living Scene?

A Living Scene is a set of nonplayer characters (NPCs) interacting with each other and with the players. Its aim is to give a feeling of life to a virtual environment, to give the

player *a sense of being there*. To achieve this, the behaviors resulting from the Living Scene have to make sense in the context of the current setting, location, story, and actions of the player. Therefore, the living scene is located in the virtual environment, situated in time and reactive to the presence of the player.

Each NPC involved in a scene fulfills a *role*. Each role in the scene is defined by a set of *behaviors* that will be executed by the NPCs to achieve the scene. The behaviors assigned to the different roles are staged in order to execute the collective behavior expected for the scene.

Our aim is to split the design of ambient life in two: the individual behaviors level and the Living Scenes level. With such a distinction, game designers are able to focus separately on each level in order to create complex individual behaviors, as well as an explainable and consistent collective behavior. Furthermore, we introduce a role abstraction layer that enables reusability and allows a wide range of possible combinations.

Back to our restaurant example, we can extract three Living Scenes: the clients exhibiting behaviors related to the restaurant such as eating, talking, and drinking (Scene 1), the friends exhibiting specific reactive behaviors related to the presence of the player (Scene 2), and the waiter exhibiting an interactive behavior with the clients of the restaurant including the player (Scene 3).

In Scene 1, each NPC assumes the role of a client and has access to three individual behaviors: "eat," "talk," and "drink." These behaviors can be staged in a specific order, or chosen automatically by autonomous NPCs. In that case, the clients decide when is the best moment to drink, eat, or talk related to their current states and the global situation in Scene 1.

Scene 2 focuses on a part of Scene 1: the clients who are player's friends located at specific table in the restaurant. Each NPC in this scene assumes the same role of friend and has access to one more reactive behavior: "make a sign to the player to join them." They can also access the behaviors of a client such as "eat," "talk," and "drink."

Finally, in Scene 3, the unique NPC assumes the role of a waiter and can execute the "interact" behavior. This behavior is triggered when another NPC enters the scene (around the waiter). More precisely, when the player enters the scene, the waiter exhibits a complex interactive behavior to discuss with the player about his or her desire to drink or eat.

11.2.2 Performing Living Scenes through Smart Zones

To put a living scene into a virtual environment and to manage its execution, we propose to define a scene through a Smart Zone. A Smart Zone is a concrete representation of a Living Scene that can be located in the environment and executed in a stand-alone way to manage the lifetime of the scene. The concept of Smart Zones is inspired by *Smart Objects* [Kallmann 99], which is often used in video games for managing agent–object interaction such as in The Sims [The Sims 99].

The idea is to include the description, within Smart Zones, of all the characteristics of the Living Scenes and how the involved NPCs execute behaviors in order to play out the scene. Thus, when an NPC go into a Smart Zone, he or she has access to every characteristic of the scene, and a role is potentially assigned to the NPC, to be an actor of the scene. If required, he or she may have to execute a specific behavior according to the other NPCs playing in the scene. As Smart Objects manage the interactions between agents and objects, Smart Zones manage the individual and collective behaviors and their relation with the NPCs interacting within the zones. When the game is running, the Smart Zones will manage the

"casting" from the available and skilled NPCs. Similar approaches, applying smart events to narrative for storytelling, have been described in earlier works [Stocker 10, Shoulson 11].

This approach leads to a decentralized control of the NPCs' behaviors in dedicated zones representing the Living Scenes. This is a way to reduce the complexity of the game control: it is not necessary to control the current situation as a whole; we can just focus on each Smart Zone instantiated in the environment. Moreover, because a Smart Zone embeds all the characteristics and information essential for the execution of the Living Scene, the control of the NPCs involved in a scene is easier: it is possible to assign, stage, and execute the NPCs' behaviors directly through the zone independently of the NPCs themselves.

11.3 Smart Zones in Practice

This section describes the definition and execution details of smart zones.

11.3.1 Definition of Smart Zones by Game Designers

To develop Living Scenes, game designers define Smart Zones in the game environment. Defining a Smart Zone means filling all the characteristics essential for the execution of the related Living Scene:

- The roles to be assigned to the NPCs
- The behaviors able to perform a role
- The orchestration of the scene by defining a sequence of behaviors
- The triggers for the scene activation
- The site of the scene (position, size, and shape)

The concepts of roles and behaviors are essential for the scene execution. The role is a way to assign a specific set of behaviors to an NPC that entered in a zone. A role is the main relation between an NPC and a scene. When an NPC assumes a role, he or she executes concretely the behaviors associated with the role in the scene. A behavior is a sequence of actions executed by the NPC during a given time interval. We decided to use these concepts because they are easily understandable and accessible for game designers. Moreover, these concepts are often used in storytelling and agent design in general.

In this way, the introduction of these concepts allows the design of the Living Scene in independent steps:

- The design of the individual behaviors
- The specification of the roles (defined by a set of behaviors)
- The organization of the Living Scene through the choice of the roles dedicated to the scene

Then, the most important part of the Living Scene design is the orchestration between the NPCs' behaviors in order to obtain a coherent collective behavior. Game designers can place behaviors of the NPCs into a timeline according to the roles, which leads to a sequence of behaviors for the Living Scene (or a part of the Living Scene) and describes the triggers to manage the starting of the scene.

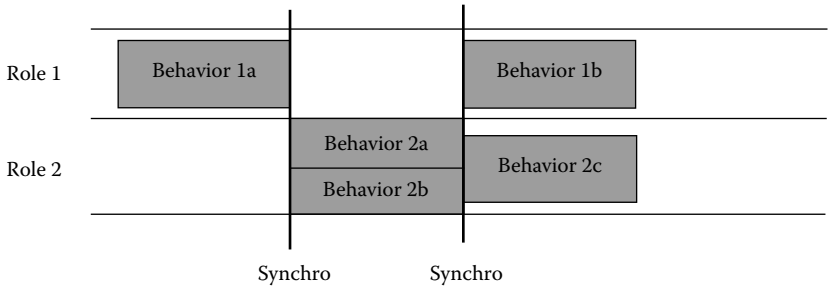

Figure 11.1

A timeline defining the orchestration of behaviors.

The timeline is organized as follows (see Figure 11.1):

- One row exists for each role.
- The behaviors are represented as boxes, and they have a beginning and an end. When several behaviors are placed at the same "time" in a single row, it means one of them will be chosen at runtime. This choice is specified by design (e.g., probability function) or automatically through the decision-making process of the NPC. This point leads to improve the variety of the observed behaviors.
- Synchronization points are used to trigger the beginning and ending of behaviors across several roles. They are added automatically after one behavior is added at the end of the timeline. However, the game designers can drag another behavior between two synchronization points to obtain specific behavioral sequences. For a given synchronization point, when all the behaviors executed by main roles end, it leads to the cancellation of the behaviors of all the other nonmain roles. For example, when the juggler finishes his or her performance, the spectators stop commenting or applauding and start congratulating (see Figure 11.6 in Section 11.4.2). If an NPC enters the Smart Zone during the execution of the Living Scene, he or she synchronizes his or her behavior with the current behavior executed by the main roles.

Finally, the game designers place the Smart Zones in the environment. The environment is considered as the "world zone" and includes all the Smart Zones. Smart Zones can overlap, in which case a priority order between the scenes must be defined. This order can be based either on the size of the zones or directly through priorities defined by the game designer.

In our example of a restaurant, each Living Scene is designed in the environment with a Smart Zone. Each Smart Zone defines the role (clients, friends, or waiter), the behaviors (drink, eat, ...), and the place of the scene in the environment. For instance, the orchestration between the behaviors in the third Smart Zone (to develop Scene 3) corresponds to a sequence of interactions between the waiter and the player in order to obtain a consistent collective behavior. The trigger of the Scene 3 consists in the entrance of an NPC in the zone. Finally, the Smart Zone to develop Scene 2 has priority on the Smart Zone to develop Scene 1.

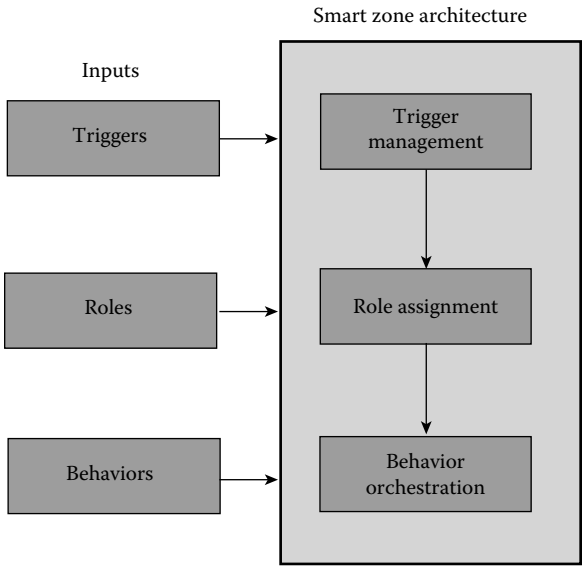

Figure 11.2

Runtime architecture of a smart zone.

11.3.2 Functioning of the Smart Zone Architecture

At runtime, when a Smart Zone is executed, it manages the lifetime of the Living Scene. In this section, we will present the runtime architecture (see Figure 11.2) for Smart Zones. It is made of several modules with different responsibilities. A Smart Zones will instantiate, start, execute, and stop a parameterized Living Scene and assign behaviors to NPCs.

11.3.2.1 Trigger Management

Triggers are a set of activation rules defining when a Living Scene should be started. Triggers can be the occurrence of a specific moment in time, the presence of a given NPC or player in a specific area, the beginning of an interaction with a particular object, or a combination of several of these rules.

This first module of the architecture aims to check whether the current situation allows the scene to be executed. The module takes as inputs the triggers defined for the scene and checks if all the trigger conditions are verified in the current situation. If that is the case, the next module steps in.

Let's consider, in our example, the waiter serving the player. The scene can start if and only if the two following trigger conditions are true: the waiter and another NPC are in the same zone at the same time, and it is after 8 pm.

11.3.2.2 Role Assignment

This module is responsible for the assignment of a role to each NPC present in the Smart Zone. An NPC has only one role in a scene. As described previously, a Smart Zone embeds a set of roles defined for the scene. In order to manage automatically the assignment of the roles, we define three subsets: main roles, supporting roles, and extras.

Main roles are essential for the execution of the scene. The scene itself revolves around characters fulfilling these roles. One or several NPCs can take main roles; they become the main actors. The scene won't start unless all the main roles are fulfilled with characters belonging to the Smart Zone. The scene is finished when the behaviors of all the main roles are finished.

The main roles are used to lead and synchronize all the collective behaviors. Main roles are essential to execute the Living Scene. The main role can be cast in two ways: either a particular NPC is defined in the design phase as a main actor for the Living Scene or the module should have to find an NPC able to endorse the role. Thus, the module chooses an NPC from the set of NPCs located into the Smart Zone. If no NPC is able to take the role, the module starts a dynamic search operation to find an NPC able to take a main role around the Smart Zone. This search operation uses an expanded zone, which is automatically extended until the NPC is found or until the expanded zone wraps the world zone. When the NPC is cast, he or she moves automatically to the Smart Zone.

Supporting roles are favorable for the execution of the Living Scene. Zero or more NPCs can take supporting roles; they become the supporting actors. They interact with the mains actors in order to execute collective behaviors. The supporting roles are first fulfilled by the NPCs present within the Smart Zone.

Finally, extra roles are optional for the realization of the Living Scene. Adding extras to a Living Scene allows the casting of numbers of nearby characters to execute "ambient" behaviors that will mostly react to the "main" set piece.

Let's illustrate these three sets of roles in a juggling show:

- The main role is the juggler; it triggers the start of the show.
- The supporting roles are the spectators.
- The extras are additional spectators as passersby, which are not mandatory for the show to be consistent, but that get involved because they are close to the show and interested from a distance.

The Role Assignment module needs to determine if a given character can fulfill a given role. This is why each created NPCs needs to be assigned a set of roles it can fulfill. The use of a role hierarchy can help to facilitate this assignment.

This role assignment step determines if the Living Scene can actually start. If at least all main roles can be cast, the runtime proceeds to the behavior orchestration. Then, each time an NPC enters the Smart Zone, the module determines if a role should be assigned to the NPC in the set of supporting or extra roles. If a supporting role is available, the role is assigned to the NPC. If all the supporting roles are fulfilled and if an extra role exists, the extra role is assigned to the NPC. If no extra role exists, the NPC does not join the Living Scene.

11.3.2.3 Behavior Orchestration

In practice, the "output" of a Living Scene is the execution of different behaviors by the involved NPCs. From the assigned role, an NPC is able to execute some behaviors defined for the role. This module is responsible for the assignment of a behavior to the NPCs, through the sequence of behaviors defined in a timeline (see Figure 11.1). The timeline orchestrates the roles of NPCs over time and synchronizes them in order to have credible individual and collective behaviors, including during a dynamic role assignment.

11.3.3 NPC Behaviors

To obtain a better ambience of life, we propose to take advantage of the principle of autonomy, by moving a part of the decision making to the NPC level.

11.3.3.1 Role Interruption

In the previous sections, we described a simple scenario where all participants of a Living Scene stay until all the main role behaviors end. The NPCs can also decide not to participate in a Living Scene according to their own goals. In this case, the Smart Zones do not entirely control the exit of the NPCs from the Living Scene, but they handle dynamic role assignments. Once more, the rules depend on the role:

- An NPC assigned to a main role can't leave the scene without stopping it if is not possible to recast the main role among other NPCs in the zone.
- When an NPC assigned to a supporting role leaves the scene, the role is cast automatically among extras and nonparticipants.
- NPCs with extra roles can leave as they wish without any incidence.

If the NPC decides to participate in a Living Scene, their behaviors are controlled by the timeline of the Living Scene. However, the selection between the NPC goals and the ones of Living Scene is based on priorities. If the priorities of goals are higher than the one of the Living Scene, the NPC can leave the scene. For example, if the hunger of a spectator is higher than its motivation of participating in the spectacle, it leaves the Living Scene and goes to eat.

11.3.3.2 Management of Overlapping Zones

As described previously, several levels of overlapping Smart Zones can result from their placement, which leads to a priority order of the scenes. The order can be defined through the relative sizes of the zones or directly by the game designers as a priority in the properties of the zones. These priorities are used by the NPCs to choose the best behaviors when they are located in several zones at the same time.

By default, the NPC chooses the behavior associated with the zone with the highest priority, but the NPC can also decide to execute a behavior from a zone with a lower priority for specific reasons, for instance:

- The behavior allows the NPC to achieve several Living Scenes at the same time
- The current internal state of the NPC allows him or her to execute the behavior but not the one from the zone with a higher priority
- The behavior allows the NPC to complete an individual goal

With such a degree of autonomy, an NPC can try to find a compromise between the roles of a scene and his or her goals.

11.4 Concrete Example

This section walks through an example scenario.

11.4.1 Scenario and the Smart Zones

The aim of this scenario example is to demonstrate how to create ambient life in a shopping street using multiple Living Scenes: "queue at the cash machine" (LS1), "spend time on a bench" (LS2), "buy interesting things at the shop" (LS3), "wait for the bus" (LS4), and "juggling show" (LS5).

The world zone is corresponding to the street, in which we define five specific Smart Zones to represent and manage these Living Scenes (see Figure 11.3):

- Smart Zone 1 (SZ1) is placed around the cash machine and triggered when an NPC enters the zone. The main role is the "cash taker" associated with the behavior "take cash" accessible when the NPC is the first in the queue. The supporting role is "queued" associated with the behavior of "wait for my turn." Several NPCs can take this role at runtime. No extra role is defined.
- Smart Zone 2 (SZ2) is placed around the bench and triggered when an NPC enters the zone. The main role is "dreamer" associated with the behavior "spend time." Several NPCs can take this role at runtime (depending on the size of the bench). No supporting or extra roles are defined.
- Smart Zone 3 (SZ3) is placed around the shop and triggered when the scenario is started. The main role is "merchant" associated with the behavior "sell things." The supporting role is "buyer" associated with the behaviors of "choose items," "buy items," and "wait for my turn." Several NPCs can take this role at runtime. No extra role is defined.
- Smart Zone 4 (SZ4) is placed around the bus station and triggered when an NPC enters the zone. The main role is "passenger" associated with the behaviors "buy a ticket," "wait for my turn," and "wait for the bus." Several NPCs can take this role at runtime. No supporting and extra role is defined.
- Smart Zone 5 (SZ5) is placed in the middle of the street and triggered every two hours. The main role is "juggler" associated with the behaviors "announce," "juggle," and "say goodbye." Only one NPC can take this role. The juggler NPC

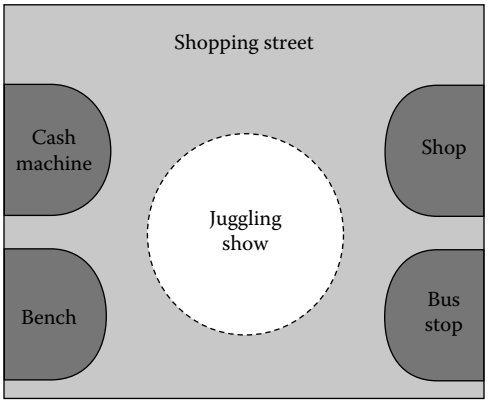

Figure 11.3

Schematic representations of the smart zones in our example.

is statically cast by the game designer before the beginning of the game. The supporting role is "spectator" associated with the behaviors "comment," "applaud," and "congratulate." Several NPCs can take this role. The extra role is "passerby" associated with the behavior "look from a distance."

By default, the NPCs that are in the world zone, take a default role of "wanderer," and can follow the default behavior of "wander" or "say hello" depending on the current situation. Thus, by default, they navigate randomly within the world zone from an activated Smart Zone to another (SZ1, SZ2, SZ3, and SZ4). To add some interaction between NPCs and the player, when an NPC meets another NPC, the behavior "say hello" can be executed.

When the first NPC enters, for example, SZ1, he or she is cast as a main role and then takes cash. Otherwise, the NPC entering in the zone is cast as a supporting role and stands in line in front of the cash machine. He or she takes cash when it is his or her turn (he or she is cast as the new main role). Similar operations occur for the other Smart Zones with the corresponding roles and behaviors.

11.4.2 Implementation in Unity3D

We have implemented the Living Scenes of the shopping street in Unity3D [Unity 14] and created the behaviors under MASA LIFE [MASA 14] using behavior trees [Champandard 08, Champandard 13]. We focus only on decisional behaviors and scene execution, and not on animation, navigation, and 3D design.

We created the roles and the associated behaviors for each Living Scene. The Smart Zones of the Living Scenes were placed in the environment of the shopping street (circles on Figure 11.4). We specified the characteristics of the Living Scenes through a dedicated

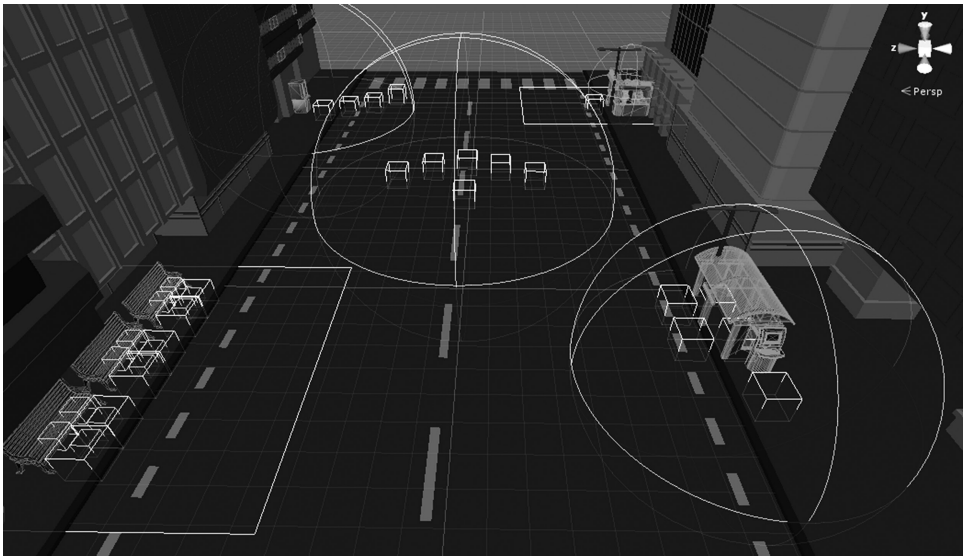

Figure 11.4

Instantiation of the shopping street example.

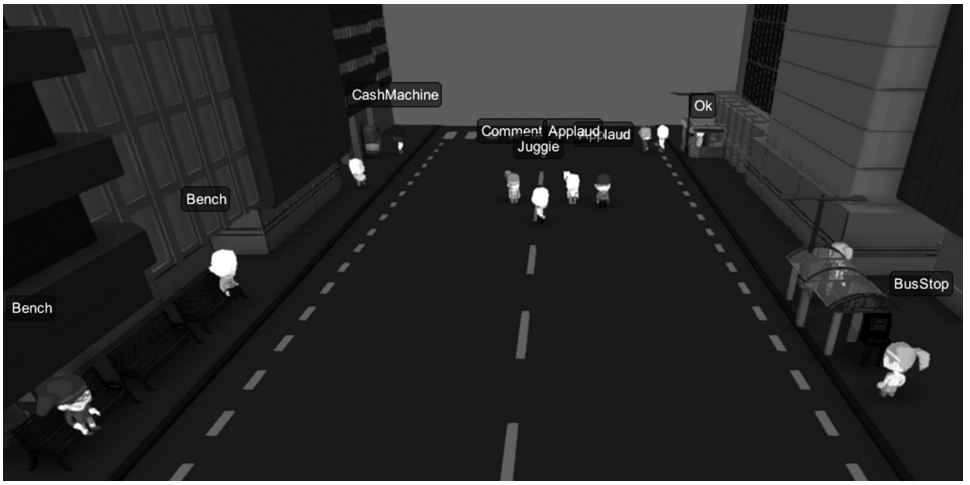

Figure 11.5

The shopping street example in Unity3D.

graphical interface in Unity3D by selecting the associated Smart Zones. Figure 11.5 illustrates the result of the Smart Zones execution in Unity3D.

Let's consider the juggling show scene. SZ5 defines five slots in which the spectators can take position to watch the show. Thus, five supporting roles are available in this specific case.

The timeline of the show defines the sequence of behaviors and their synchronization for the juggler and the spectators (see Figure 11.6). The show has three stages: announcement, juggling, and end. During the announcement, the juggler informs that it will begin the show soon (he or she executes the "announce" behavior).

The trigger for the scene is a given period. When the time comes, the juggler is automatically cast. In our example, the designer associates a particular NPC to the role of

Figure 11.6

The timeline for the juggler spectacle.

juggler before runtime. The NPCs already in SZ5 when the scene is triggered are first cast as "spectators" if they can assume the role, until all the supporting roles are filled. The remaining NPCs cast as "extras" if they can assume the role. Moreover, NPCs interested in the show can go to SZ5 attracted by the announcement, and they are cast in the same way. They automatically adapt their behaviors to the corresponding stage of the show. During the juggling phase, the juggler juggles, and the spectators comment or applaud randomly. At the end of the show, the spectators congratulate the juggler, and the juggler says goodbye, and then the scene ends.

11.5 Conclusion

Living Scenes allow the execution of an ambience of life in video games with autonomous and less repetitive nonplayer characters exhibiting consistent, credible, and interactive behaviors. We propose a solution with Smart Zones to manage the NPCs' behaviors for the Living Scene execution and thus to move the complexity of the design into the zones. We implemented our solution into MASA LIFE and instantiated an example with Unity3D to test the Smart Zone concept, with positive results.

We demonstrated the prototype to game designers and their feedback is very encouraging. The model and the architecture seem to fit their needs to create ambience of life. Despite the simplicity of the scenario example presented in the previous section, the shopping street in Unity3D gives a good impression with less repetitive and more credible behaviors. Through this first implementation, we noticed that without the proposed model and architecture of Smart Zones to define and execute a Living Scene, the design of such an ambience would have been more complex.

This work is a part of the OCTAVIA Project, a research project funded by the French government (DGE). The main goal of the project is to propose an innovative tool to design and test scenes of ambient life involving interactive and autonomous nonplayer character in 3D environment. In this project, we plan to evaluate our solution design in order to validate its usability and whether our solution concretely simplifies the design of ambient life.

Although Smart Zones are an interesting and accessible approach to design Living Scenes, two main challenges need to be addressed to complete the solution: the integration of the player into a Living Scene with the impact of his or her actions on the scene; and the simplification of the design of the behaviors of autonomous NPCs in relation to Living Scenes, especially to allow NPCs to reason about Living Scenes.

Acknowledgments

This research is funded by the French government DGE within the OCTAVIA Project (PIA-FSN-2012) and supported by the cluster Cap Digital.

References

[Champandard 08] Champandard, A.J. 2008. Getting started with decision making and control systems. In *AI Game Programming Wisdom*, Vol. 4, pp. 257–264. Charles River Media, Hingham, MA.

[Champandard 13] Champandard, A.J. and Dunstan, P. 2013. The behavior tree starter kit. In *Game AI Pro: Collected Wisdom of Game AI Professionals*. A K Peters/CRC Press, Boca Raton, FL.

[Kallmann 98] Kallmann, M. and Thalmann, D. 1998. Modeling objects for interaction tasks. In *Proceedings of the Ninth Eurographics Workshop on Animation and Simulation (EGCAS),* Lisbon, Portugal, pp. 73–86.

[MASA 14] MASA Group. 2014. MASA LIFE. http://www.masalife.net (accessed September 10, 2014).

[Shoulson 11] Shoulson, A. and Badler, N.I. 2011. Event-centric control for background agents. In *Lecture Notes in Computer Science*, Vol. 7069, pp. 193–198. Springer, Berlin, Germany.

[Stocker 10] Stocker, C., Sun, L., Huang, P., Qin, W., Allbeck, J.M., and Badler, N.I. 2010. Smart events and primed agents. In *Proceedings of the 10th International Conference on Intelligent Virtual Agents*, pp. 15–27. Springer-Verlag, Berlin, Germany.

[The Sims 99] The Sims. 1999. Electronic arts. http://www.ea.com/sims (accessed February 7, 2015).

[Unity 14] Unity Technologies. 2014. Unity 3D. https://unity3d.com/ (accessed September 10, 2014).