LEA
Leonardo Electronic Almanac
Vol 16 Issue 2 – 3

Interesting Times – Modelling Time in Multi-Player and Massively Multi-Player Role Playing Games

Anders Tychsen
Department of Computing
Macquarie University
Building E6A
2109 North Ryde, Sydney, NSW
Australia
Email: anderstychsen [at] yahoo [dot] com

Michael Hitchens
Department of Computing
Building E6A
Macquarie University
NSW 2109
Australia
Email: michaelh [at] ics [dot] mq [dot] edu [dot] au

Keywords
Multi-Player, role-playing, computer games

Abstract
Time is a key concept in the design, playing and study of games and can be viewed from multiple perspectives, e.g. the player and the game world. Here a comprehensive, iteratively developed model of game time, based on empirical games research as well as recent theory, is presented. The model is tested in practice and its applicability across tabletop, digital and other forms and formats is demonstrated. Special emphasis is placed on multi-player and massively multi-player games, as well as role playing games, as these feature complex game time behavior that cannot be explored in existing models of game time. The model includes seven viewpoints of game time, and allows for the mapping of time as an interactively created and non-linear feature of games and gameplay.

Introduction

Computer games are inherently linked with the concept of time. It takes time to play a game, time has a definite place in our understanding of the gaming activity and time is an important factor in the design of computer- as well as tabletop games. From calculating reload times of weapons in FPS games to achieving perfect balance, level and story design, pacing of play, timing of events and triggers, time is a major element in the development and design of games [e.g. 2,4,17]. Time is furthermore a key consideration

to address in games research, as it is one means of understanding what happens in a computer game, how the game and the player(s) interact, as well as how the game is experienced by the player(s).

While an important aspect of play and design, there have been few contributions towards developing a comprehensive theory of time in computer games [10,11]. What efforts have been made draw from a variety of sources, including semiotics [13], physics [15], literary narrative models [8], and user-game interaction [9,10,11]. In contrast, time is a subject well studied in other academic areas of study, such as film, literature and theater. Given the importance of time in critical understanding in these areas and its obvious importance in relation to games the need for a clearer understanding of time for the study of game should be obvious.

In creating a theory of game time, it is important to realise that there are differences between time in games involving one player, and those involving more than one player. In a single-player game, only two entities are involved, the player and the game. In a multi-player game, the actions of one player can affect the game playing experience and actions of the others, or players can be engaged in different activities while the game operates. This creates a complexity that a model of time from a single player perspective will not encompass [9]. Most of the contributions mentioned above pay scant, if any, attention to the inclusion of multiple players in their models.

One of the most popular categories of multiplayer game is Role Playing Games (RPGs). RPGs span a huge range of formats, from tabletop, virtual reality, mobile-based and online. Despite the common denominator – role-playing – there is an incredibly large variety of formats employed, and it is therefore challenging to describe and analyse these games in the context of a coherent framework.

As a shared – if highly variable – feature of all game forms, game time offers a venue for studying and analysing RPGs across different formats. A coherent theory and model of game time would for example be useful in modelling player interactions in RPGs, and the collaborative storytelling process of Pen and Paper RPGs (PnPs).

RPGs often demonstrate very intricate temporal behaviours. Many forms of RPG present players with a high degree of freedom, allowing them to operate independently, and affect the game story. Furthermore, RPGs are based on detailing the activities of player characters (avatar-characters in computer games) – activities that do not need to follow the chronological flow of time in the fictional worlds Player characters may exist in different instances of the game world chronology, or even form personal perceptions of time that operate completely or partially in the mind of the player. Briefly, multiple players and their interactions create a comparatively larger degree of complexity when compared to single player games. This means that a model of time developed for such games, while not necessarily universal, should form a basis for a wider understanding of the nature of time in games.

The purpose of this paper is to provide a model of how time operates and flows in multi-player (MP) and massively multiplayer (MMP) games, and apply it to the four major formats of RPGs – the traditional tabletop PnPs, Computer-based RPGs (CRPGs), Massively Multiplayer Online RPGs (MMORPGs) and the physically embodied formats collectively referred to as Live Action RPGs (LARPs). Focus is here on digital MP and MMP games, notably RPGs, however, the model is equally applicable to non-digital games and single-

player games, and several examples from e.g. tabletop games have been included in the model description.

The model presented expands the single-player model presented by [9], which is itself an extension of [10,11], into the MP and MMP range. The model is not intended to be the final word on game time,

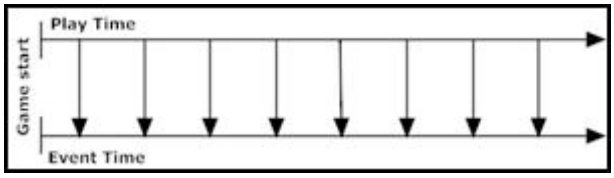

Figure 1: Mapping player actions between two layers of time on a 1:1 basis (after [10]).

however it benefits from being scaleable and adaptable to specific purposes. For example, the use of segmentation of temporal activities integrates the interactive nature of games and allows different levels of granularity in the analysis of time in games to be utilised. While previous models are based on theory and/or observations of games, the model presented here is derived from theory, play testing and empirical research. Multi-player PnP and CRPG experiments were employed in order to test and refine the game time model and provide a test bed for the application of the model to RPGs in practice.

Previous Work

Discussion of time and its related issues has featured in numerous computer game design books as well as games studies publications [e.g. 1,2,4,13,14,16,17] Games have even been classified based on their use of time, e.g. real-time strategy, turn-based games. Time is integrated in game design terminology within computer games as well as other game formats, e.g. the concept of game speed, and the utilisation of direct time control (e.g. the rewind function of *Prince of Persia: The Sands of Time*, or bullet time as used in *Max Payne*). However, there have been very few dedicated studies of time in computer games from the research community.

[8] Adapted concepts from narrative theory for the description of temporally related phenomena in games, noting that events can be described in terms of their duration, ordering, speed, frequency, simultaneity and the time of action. [8] also noted that the dominant temporal relation in computer games is between the time of the user and the events of the game. This in opposition to the temporal relation observed in traditional narratives between the discourse time (the time of the telling of the story) and the story time (the time of the events narrated).

[15] Discussed time from a physics point of view, e.g. separating relative and reversible time. [5] Considered time in computer games from an aesthetic perspective.

[13] Outlined a semiotic approach based in literature studies. [13] Created a framework for ludic semiotics based on the recognition of different temporal semiotic systems: simulation, game and narrative. As this approach is focused on structuring events when designing a game, rather than the actual experience of time when playing a game, it cannot be directly compared with the approach adopted here.

[10,11] (Figure 1) presented the possibly first model for game time from the viewpoint of the playing experience, rather than game design. That model incorporates time in the form of layers: **Play Time** – the actual real-world time spent playing - and **Event Time** – the time of the fictional world of the game - and describes the ways that the game and the player can interact via these two layers. The model allows the **mapping** of time between the two layers, i.e. projecting the play time and the actions of the player onto the event time layer. This approach allows a description of game speed (the relation between play time and event time). [10,11] describes game features such as cut scenes, loading and saving as specific effects that can occur in the play time-event time relationship.

[9] Adopted the approach outlined by [10,11] (Figure 2), refining the model to include **non-linearity** [1], expressed as a branching of certain time layers to represent the ability of players to choose between different paths, and to abandon their current path and return to an earlier point in the game. Non-linearity in the model allows the integration of the potentially complex relationships between the player's experience of time in the real world and progress through the game. For example, by reloading a previously saved game, the player is effectively turning back the event time of [10]. Non-linearity is relevant to all games because all games feature choice. This even includes PnPs, as those games permit the rolling back/forward of the game story, should the participants agree.

The model of [9] also introduces multiple **modes of engagement**, by dividing the temporal layers up into **segments** and **breaks** (Figure 2). Segments represent a section of time of the relevant temporal layer, within which the activity is homogenous - for example, waiting while a module is being loaded in the CRPG *Neverwinter Nights.* Breaks represent changes in the activity, and have no temporal dimension, being instantaneous. For example, starting to play (interact with the game) after the loading of the *Neverwinter Nights* module finishes. Breaks should not be confused with periods of inactivity, i.e. a period where the player is not playing the game. These have a non-zero distribution in the relevant temporal layer.

The model of [9] has four time layers:

1) Playing Time: The objective real-world chronological time experienced by a player during and between game play sessions.

2) Engine Time: The objective chronological real world time in which a game engine executes.

3) **Progress Time:** An abstract (non-chronological) measure of time tracking movement towards game completion and allowing events to be related in terms of happens-before and happens-after. This layer allows non-linearity to be described and modeled in terms of the player's achievement of the game-play aims.

4) **World Time:** Chronological or abstract and logical (depending on the game) time within the game world. While this is comparable with the event time layer of [10,11], the latter does not include the non-linearity included in the view of the game world time.
Like the model of [10,11], activities within one layer can be mapped to another layer. The first two layers are linear, while the second two represent views of time that can be non-linear (Figure 2).

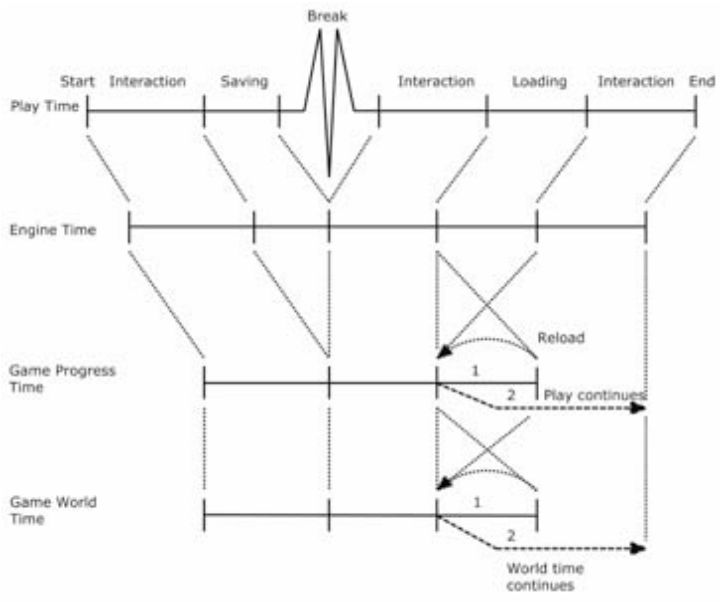

Figure 2: *The four layered model of [9]. Segments are separated with a vertical line. The dotted lines represent mapping of segments between layers. Branching occurs in the two lower layers when a saved game is re-loaded (dotted arrows) and play continues along a different path (stippled arrows). Note that while time passes in Playing Time and Engine Time (marked as a segment), the reloading is instantaneous in Game Progress Time and Game World Time (marked as a break).*

The model presented here represents a refinement of the [9,10,11] models to include **multiple players**, incorporating further theory and empirically derived data, as well as extending the model to include the complex temporal relationships of multi-player RPGs.

Method

A core aim in developing the model for game time presented here was to ensure that it was directly applicable to the study of games, rather than simply an abstract theory. An empirical approach provides support for the theory behind the model and demonstrates its applicability by evaluating its ability to function in a real world context, as well as providing the hard data needed to analyse how game time operates in RPGs.

Earlier work on game time – both within games research and design – has shown that game time is a varied feature. Prior work on RPGs [7,17,18,19,20,21] indicates that time in these games can form complex constructs, with e.g. player characters operating at

different intervals of the chronological time of the fictional game worlds that these games are set in. In order to develop a model that is not only applicable but also broad enough to encompass the different features and viewpoints of game time, theoretical models were combined with a thorough, case-based study in an empirical framework (experimentation and observation).

The empirical part of this study was focused on PnPs and CRPGs. PnP and CRPGs include complex manipulations of game time, due to the high degree of player freedom as well as the unrestricted communication between the participants [7,21]. This was combined with extensive MMORPG play testing to supplement the available research and design literature, notably with *World of Warcraft*; in order to assess the applicability of the model to MMORPGs, and to expand the description of time in these games. The practical difficulties in observing the large number of participants over a large geographical area precluded LARPS from being included in the experiments, however, all available theory on LARPs was included in the study [e.g. 3].

The model was developed and refined over the course of two rounds of empirical experiments. A basic assumption is that the experimental situation is representative for PnP and CRPG games. This is potentially problematic, because the variation in PnP and CRPGs [7,20,21], makes establishing experimental conditions that are standardized difficult at best. This assumption is alleviated by combining the empirical experiments with literature studies, which serves to locate any game format variations that would impact on the game time model. Furthermore, by using time layers as the foundational building block of the model, it becomes resistant to error introduced by minor game design variations (for example, the number of players or distribution of directorial control). For example, whether the PnP in question utilises a GM or not does not impact on the model framework, only on the application of the model to the specific game situation.

The CRPG *Neverwinter Nights* used in the experiments is a fairly typical example of the CRPG genre, and examination/playtesting of a range of other CRPGs such as the *Might and Magic* series, *Baldur's Gate, Sacred*, *Diablo* and *Summoner* revealed no issues in relation to the model that needed to be integrated. Undoubtedly, with the huge variation of games, there are aspects that have not been captured in the current model, and it must be stressed that the model presented here is not meant to be the final world on game time.

The PnP and CRPG experiment setups were designed to provide as similar conditions as possible. This is in order to limit bias caused by differences in story, rules, players etc. between the two game formats. During the pilot test, one multi-player PnP and one multi-player CRPG game session was run. The PnP session consisted of five players and one GM, while the CRPG session consisted of the same players interacting via and with the game engine. The same group of players played both game sessions. The modules being played were based on the same widely used rules system; the D20 system originally designed for the *Dungeons & Dragons* PnP, version 3.5 of which forms the rules basis of the CRPG used, *Neverwinter Nights*. Each player had his/her own computer monitor.

In order to limit the number of variables involved, the two modules both featured story themes of reversal and revenge. In both game sessions, the players were situated around a table with full verbal and visual communication access. The game sessions were recorded on video. For the CRPG, a client tool bundled with *Neverwinter Nights* was utilised to capture screen-based actions. For both game sessions, selected segments of the

player communication was transcribed and coded, and for the CRPG a game log extracted showing all text-based chat as well as the use of emotes. The players were asked a series of questions before and after the game sessions. This was both to locate any tensions in the groups that could bias results, to obtain information about how players relate to their fictional characters in the different layers of time proposed in the models of [9,10,11].

The 2nd round of experiments consisted of a series of 10 PnP and 10 CRPG sessions, each involving 5 players (plus 1 GM in the PnP sessions); running between 3-7 hours. All game participants

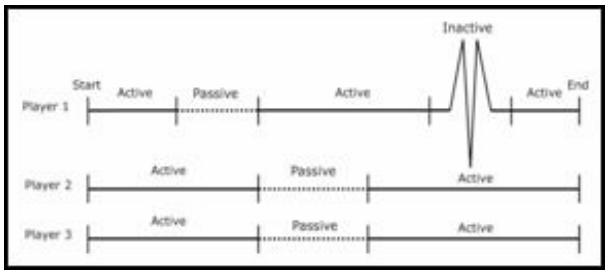

Figure 3: An example of the Playing Time of a three-player game, where the players change states. Note the period of inactivity of Player 1. The game however continues because the other two players are active.

were adults (18-54 years of age, 28.8 average, 27 median). The groups carried over between the two game setups.

The recordings, questionnaires and game logs formed the basis for iteratively revising the game time model, and if the model encompassed the full variety of player actions. A substantial amount of revision was necessary following the experiments, for example, the need to separate game progress time into two distinct layers (game progress and -story time), was realized after the first round of experiments. The experiments also gave rise to the boundary between the Engine Time and Server Time layers being redefined (see below); and gave rise to the notion of perceived time. Finally, examples of the different time layers in operation were located to serve as documentation.

Key Concepts

Before describing the current game time model, a few key concepts need to be introduced and defined. These concepts are used to describe how game time operates and how players interact with game time.

Player state: The player state describes the situation of the player for a given segment of time along any of the seven time layers. Three general states are defined:

Active: The player is actively playing the game, interacting with it and/or the other players (Figure 3).

Passive: The player is playing the game but is currently in an observatory mode, i.e. not actively involved in the game play. For example, a goal keeper in a soccer match when the ball is in the other half of the field, a player in a PnP who is watching the GM and two other players acting out a scene; or auto-crafting in MMORPGs.

Inactive: The player is not playing the game, for example, during the real-world time between a save and exit; or during the start and load of a computer game. During inactive periods, neither the player nor any avatars are active in the game.

The identification of player states is not strictly necessary in order to model time flow in MP and MMP games, however, it is of relevance when considering how multiple player can affect the game time of each other, and how the game utilises time. This is of special interest to PnPs, where the players constantly change between active and passive states depending on whether or not their characters are actively involved in a given interaction.

Segment granularity: By dividing the layers of game time into segments of a defined duration, separated by breaks, a useful tool for mapping different types of player activities is gained. However, it is obviously possible to use different levels of granularity when defining the segments of a game time layer (Figure 4). For example, while actively playing a game, a timeline segment can be labelled simply 'interaction'. However, in a more detailed study of the player activity it might be of relevance to apply a finer grain. Irrespective, the level of detail must be tailored to the specific game analysis, with sections labelled according to the specific task the player is performing within the segment (e.g. actions such as loading and saving can occur in different ways and have varying impact, e.g. conditional saving in *Animal Crossing* or server save/load 'roll-backs' in MMOGs).

When applying the model presented here to the recorded game sessions, it was observed that in general, the finer the grain of the segmentation applied, the more variation there was in the activity of the different players within each segment. Note that in the models of [9,10,11], individual time lines can potentially represent any number of entities. When considering interactions in MP and MMP games, the various time lines can be used as representations of specific players or groups of players (Figure 5). In this case, one time line represents the specific group or guild of players.

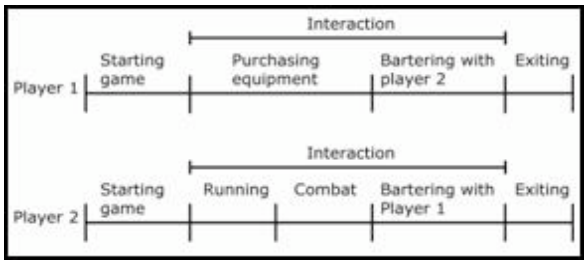

Figure 4: Example of how the activities of players can vary within the same Playing Time segments in multiplayer games. Note the two levels of segment granularity in the centre of the two time lines (the

'interaction' segment is broken down into segments of a smaller granularity).

Game Speed: [10,11] defined the ratio of time between his play time and event time layers as the game speed. While this is a useful definition when applying his dual layer model to game study, the current model utilizes additional layers, meaning that defining game speed based on only two layers becomes impractical. The term relative game speed is therefore utilized here in order to denote the ratio between the time flow of two specific time layers. For example, the relative speed (or ratio of time flow – the progression of time within a given framework - difference) between Playing Time and World Time in a CRPG like *Neverwinter Nights* is 1:12 – *Neverwinter Nights* operate on a two-hour day cycle determined by the active Playing Time.

The Seven Layer Model

The model of time presented here is an extension of that presented in [9]. That model is extended here to account for the presence of multiple players and the effect that has on the nature and perception of time in multi-player games. The presence of multiple players means that the individual player not only interacts with the game, however usually also with other players. This impact might be limited to specific layers of time or effect all of them and the magnitude of this effect can vary tremendously between different MP and MMP games.

The model includes seven layers or viewpoints of time. Four of these are adapted from [9]: playing time, engine time, progress time and world time. A substantial amount of new detail is added to accommodate the multiplayer situation, as described in the following sections. The three new layers (server time, story time and perceived time) are introduced to accommodate the presence of multiple players. These new layers are necessary in order to describe game time operation and the way multiple players can interact with each other and the games they play. Note that not all seven layers are applicable to all forms of games, e.g. PnPs and other table top games do not have Server Time and Engine Time layers, as these are bound to the hardware and software of computer gaming. The model is intended as a toolkit for the understanding of games and it is expected that not all layers would be necessary in all applications

Playing Time

Playing Time (Figure 4) is a conceptually straight forward measure of game time from the viewpoint of the player. It is defined as the objective real world chronological time experienced by a player during and between game play sessions. Playing Time has a specified beginning and end, between which game playing occurs. The playing time includes all the time the player interacts with or otherwise participates in the gaming activity within the magic circle of [17]. The core of the time spent playing MPGs and MMPGs is taken up by performing game related actions and interacting with the other players, via various input devices – verbal, game pieces, human-computer interaction devices such as mouse and keyboard, etc. Via these actions the players affect the game state, progressing through the game. Each player (or group) would have their own separate representation of playing time when modelling a game.

The presence of multiple players means that multiple different actions can be taken in the game world at the same time by the different players. For example, a player can choose to attack a MOB while another player opens a treasure chest. This means that within a given time segment, players can perform different activities. Some of these, for example bartering or negotiating, can temporarily align the activity of the players, as seen in Figure 4.

Engine Time

Engine Time considers time from the perspective of the actual game engine or software running the game application, and can be used to map e.g. loading, saving, briefing and interaction segments. It forms in computer games one of the two aspects that the player can interact with – the second being the other players.

This time layer is defined by the objective, linear and chronological real-world time in which the client software part of the game engine executes, and can be fairly similar to Playing Time, however, there are subtle differences (Figure 2). For example, in Engine Time there is no measure of time flow when the game engine is not operating, as the software is not functioning during that time (Figure 2, the break in the Playing Time does not register in the Engine Time layer). In multi-player games engine time refers to time as experienced by the client software for a particular player. The client software for each player in a multi-player game will experience time and events in a potentially different fashion, due to the nature of distributed computing systems.

Engine time is also used for the software in multi-player ("hot-seated" games that run on a single computer). The concept can also be applied to board games such as *Monopoly, Axis & Allies* and *Chess*, or PnP games. In these cases Engine Time represents the viewpoint of the game itself, and is mapped to Playing Time in a similar fashion as for computer games.

Server Time

For single player games the game software is normally located on a single machine and time from its point of view can be represented via engine time. Multi-player games are often implemented using the client/server paradigm. Each player has a copy of the client software on their machine, with a single server (possibly on the same machine as one of the clients) providing central control. The server has its own viewpoint on time, which is normally different (at least in detail) from that of any of the clients. This is especially true in MMORPGs, where the server can remain running for extended periods, even if no clients are operating.

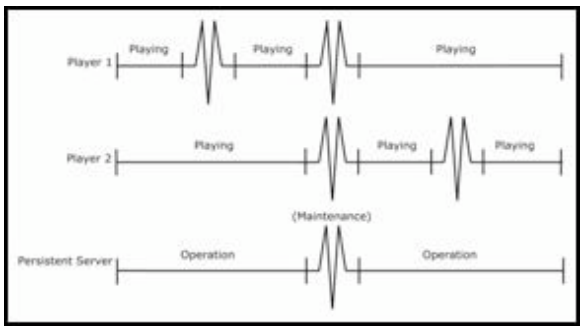

Figure 5: Multiple players of a computer game with a persistent server. The top two timelines are Playing Times for Player 1 and 2, while the bottom timeline represents time as viewed from the perspective of the persistent server, i.e. the Server Time.

Server Time (Figure 5) is here defined as the chronological, real-world time in which a game server hosting the server-side software executes. Software on separate machines does not share the same view of time [12] and this requires a separate layer of time in multi-player games. In modelling multi-player games not implemented via client/server software (e.g., as peer-to-peer software) this layer is not required, although such architectures are very rare in games software and the remainder of this section refers to games implemented using the client-server architecture. Server Time is useful when mapping differences in player activities in the Playing Time layer to the game itself in a MP or MMP situation. Because each player has their own Engine Time layer, the Server Time layer allows the activities of the players to be reconciled. In the case of MP games such as *Counterstrike*, where the computer of one player can acts as both the server for the game as well as hosting the game engine for that player, the game engine time of that player is mapped directly to the server time. In general, precise synchronization between computers is nigh impossible due to network lag and processing time, which causes variations in when each instance of a digital game receives information about events in the others. This imprecision can be important a game analysis, depending on the segment granularity employed.

In persistent-world games, the Server Time layer is almost continually active, save for occasional down times for maintenance or similar issues. Players are free to log on and off as they choose, leading to a complex mapping between player and server time (Figure 5). While in a single-player computer game the Engine Time maps to part of the Playing Time (Figure 2), the situation in a MMORPG is reversed. Even when players are inactive, the server continues, even if the player avatar is effectively removed from the game (some MMORPGs allow certain repetitive activities such as crafting or transportation without the player needing to actively interact with the game). The gap in Playing Time is therefore mapped to a segment in Server Time, as opposed to a break in Engine Time.

In summary, for each MP or MMP there will be one Server Time layer, mapping to multiple Playing Time and Engine Time layers. Note that e.g. MMORPGs can have multiple shards, or copies of the game running simultaneous, however these can be viewed as separate games or systems.

Progress Time

While Playing, Engine and Server times are all linear and chronological, games can provide a non-linear experience [1,9,17]. Non-linear can have multiple meanings in a

games context, one being that there is more than one possible path from the beginning to the end of the game, referring to the topology of the nodes which make up the game structure. A game which has multiple paths to completion is nonlinear in this sense and the choices of the player when navigating the game impacts on the experience of the game and game world. For example, in a game of *Chess*, a player will usually have more than one option in making the next move.

Considering game time from the point of view of the player, rather than the game (world), non-linear can also refer to the ability of players in most digital games to choose between paths, and later abandon or revisit paths. For example, when reloading a saved computer game and replaying a specific section, thus experiencing the same content. The replayed section will generally be different from the originally played (Figure 2), as the player has knowledge of the path being replayed. This therefore leads to a branching in the game time (Figure 2, 6), as a similar amount of game World Time is experienced, but at a different point in Playing Time and normally with different player choices and outcomes, although as noted by [9], mistakes can be repeated. Branching can also occur when progress can be lost without reload, for example in strategy board games such as *Twilight Emperium* and *Settlers of Catarn*, as well as many card games, the goal of players is to reach a certain amount of points. These are scored by reaching specific conditions, such as controlling a specific planet in *Twilight Emperium.* In such games, players can however loose points already gained. In terms of game progress, this type of setback reduces the player to a previous checkpoint (one with fewer points), causing a branch to occur.

Progress time is a logical measure of a player's progress through the game. It can be measured objectively by mapping the specific game events from the beginning to the end of the game, and organizing them into logical relations of happens-before and happens-after, for example, by using a series of game objectives or checkpoints in the game story. Progress time can be measured based on different types of criteria, depending on the game in question. Two primary forms of progress observed in the empirical experiments for this study **mechanic progress** and **task progress**. Progress time allows the mapping of branches caused by e.g. a save-reload cycle (Figure 2). While different branches in progress time can represent similar intervals of world time, they have no effect on the future of game except for that the player has learned about the replayed section of the game. Also note that in a multi-player game there will be a game progress time measure for each player. Mechanic progress changes the game state in terms of the rules, e.g. acquiring a character level in a MMORPG such as *Ultima Online* or the accumulation of money in *Monopoly*. Progress in these terms can be lost via a reload, and non-linearity introduced. In tabletop games this is possible where the basis of the measurement can be lost and regained. Task progress is related to the requirement of the players having to complete certain tasks (objectives, quests, etc.) to advance in a game. Examples of this include the levels of games such as the classical *Pac Man* and *Atomic Bomberman*. Note that some games (e.g. many PnPs) have no defined completion conditions; however the progress of players from one point in the game to another can still be readily observed. This approach is useful in games that do not enable players to revisit previously explored paths, i.e. most non-digital games, in which Progress Time would otherwise be linear and possibly chronological.

Story Time

Story Time is time of the actual dramatic story of the game, if present. Story Time is chronological if the game World Time is chronological, and otherwise logical. However, most games without chronological world time have limited story, so Story Time is most useful as a chronological measure. Following the same general principles as Progress Time, Story Time can be non-linear (Figure 7). Story Time tracks the movement towards game completion according to the game story, allowing events to be organized in orders other than the strict chronology of the game world (eg., via foreshadowing or flashbacks). The term "game story" is here used in a loose sense, as the concept of game narratives is a contested issue [11,13]. However, irrespectively of whether or not game-based stories or plotlines fulfil the formal criteria for being narratives in the traditional literary sense [11], stories form an important component in many genres of games, notably computer game genres such as adventure games (e.g. *Beyond Good and Evil, Gabriel Knight, Indiana Jones and the Infernal Machine)*, FPS games (e.g. *Deus Ex, Chronicles of Riddick, Half-Life*), and especially in all forms of RPGs.

In Story Time is included most of the narrative tools used in storytelling such as flashbacks, (direct) foreshadowing and visions, contracted time (e.g. *Civilization* and similar turn-based games) or extended time (e.g. the bullet time of *Max Payne*), and the use of these time-based narrative tools impact on the relationship between time layers (Figure 6). Story time and progress time are closely related, but are not simply the logical and chronological sides of the same coin. Story time can measure elements (such as optional quests) which may not be represented in progress time. Logical progress towards game completion, such as obtaining character levels and choosing the new abilities that flow from them, may involve no story time (although the activities that earn them may). Viewing game progress from the point of view of the Story Time can be useful, e.g. in the planning of story-based vs. mechanics-based rewards, or to measure the rate with which players progress through the game story, locate "dead" story elements which the players ignore or get stuck in, etc.

World Time

As mentioned above, World Time is the chronological or abstract time within the game world, and mirrors the non-linearity of game progress rather than proceeding linearly (Figure 6). World Time is undefined if no game world exists, e.g. traditional card games such as *Poker, Blackjack* and *Bridge.* World Time is possibly the most variable layer or viewpoint of game time. Abstract and logical World Time is used in e.g. the classic computer games *Pac Man, Atomic Bomberman* and *Space Invaders*. The players are not informed about how long a given action in the game takes in the fictional world of these games. Firing a shot at the invading aliens in *Space Invaders* could take a second or a minute in terms of World Time, this is not known. However, the World Time of these games can still be mapped logically. This leads to a complex mapping between World Time and Playing Time. In some computer game genres, notably CRPGs, FPS' and adventure games, the time flow of the game world is commonly to synchronous with real-world time, however, the day/night cycles can be substantially faster; frozen (e.g. the never-changing gloom of *Quake III* or *Hexen*) or without any discernible order beyond acting as ambience (e.g. *Serious Sam*). For example, in the later installments of the CRPG *Might & Magic*-series, a game could last years in terms of the World Time, but a few dozen hours

of Playing Time. In this example, World Time operates at more than one level (world time as measured by the in-game calendar and as apparently occupied by in-game actions) and in mapping to other layers of time a decision must be made about which World Time level/-s to map from. While World Time in PnPs can vary immensely from game to game, they are generally associated with fictional worlds where time flow is chronological and asynchronous with real-world time (e.g. player characters can travel between distant locations in second of Playing Time).

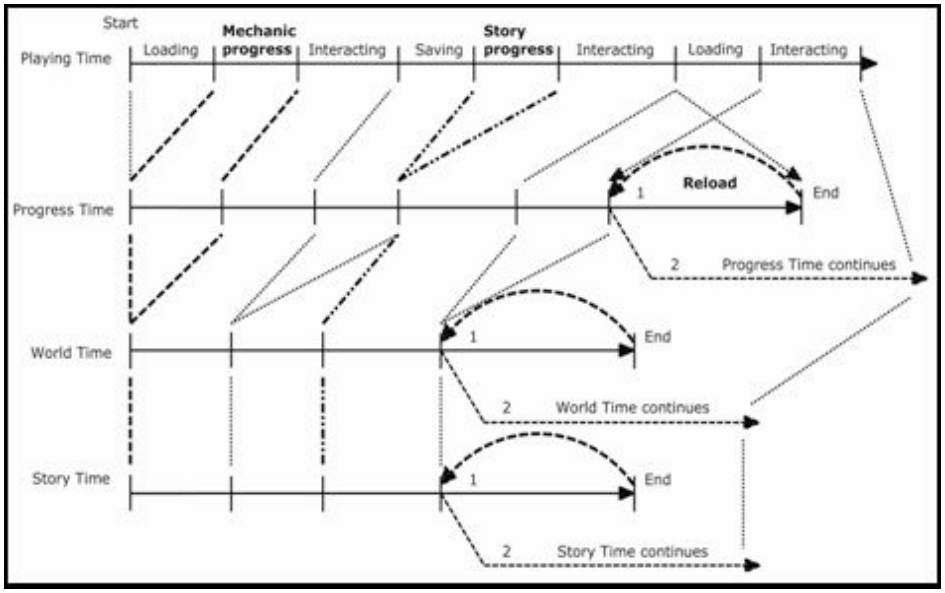

Figure 6: Mapping between Playing Time and the three non-linear layers (Progress, World and Story Time). Dashed lines: 'Mechanic' progress mapping, dashed/dotted line: 'Story' progress mapping, dotted lines: Indicates mapping between layers, bold dashed arrows: Reload, thin dashed arrows: Branch continuation.