Structuring Gametime

If the perception of time is understood as the perception of change given by a succession of events, structuring gametime entails organizing these events into particular sequences. This section will focus on technical aspects, design features, and general principles that give rise to and structure a video game's temporality. These elements will be classified into three categories: *change of state*, *space-time*, and *conditions*. Each category of this three-part typology contains a number of different features, which altogether constitute a toolkit for the analysis of gametime (table 1.4). This breakdown of the basic temporal elements of video games should be helpful for game scholars, game designers, and anyone who wishes to better understand how events in video games are portrayed and sequenced.

Table 1.4: Typology of temporal structures in video games.

Change of State	**Space-Time**	**Conditions**
Events	*Navigation*	*Time Gauges*
Pace	*Location*	*Turns*
Cycles	*Space-Time Obstacles*	*Progression*
Pause	*Triggers*	*Objectives*
Layers	*Stages*	
Reset		

The first category, *change of state,* focuses on the capacity of computers to simulate moving entities and display them through output peripherals like screens and speakers. The category examines how gametime arises from changes on the audiovisual layer. Each change of state is considered an event, and these can be manipulated by, for example, pausing or rewinding them, or varying their pace.

Space-time, the second category, analyzes the deep connection between space and time. Through the design of space and the placement of entities in it, developers can structure sequences of events and alter the player's experience of time. The final category, *conditions*, describes how time can be structured by limiting and directing the player's behavior within the game. This can be achieved, for example, by setting objectives and applying restrictions such as countdowns.[1]

From the perspective of a game scholar, this typology can be applied to the analysis of any video game's temporality. By playing a game with the three categories and their respective features in mind, it should be possible to observe which features in the game play a role in structuring events.

Game designers can use the typology to think about the temporal structures of the game they wish to design. The numerous features that constitute the typology could be seen as a list of ingredients that can be used and modified in the design of a video game's temporality. In this sense, they are what Salen and Zimmerman call *game design fundamentals,* which "form a system of building blocks that game designers arrange and rearrange in every game they create" (2004, p. 7).

This typology also aims to establish a language for later parts of this study. Consequently, the subsequent sections that lay particular focus on any of the described features will be referenced when pertinent.

Finally, it should be noted that the elements discussed in this section are not regarded as necessary or essential, but rather as recurrent observable properties of video games. This list is crafted to be a comprehensive toolkit for analysis, but it is surely not exhaustive. Even if it were, it would probably not remain that way for long considering the fast-paced evolution of the medium. Thus, this typology is permanently open for revision.

1 This typology is informed by research conducted by members of the Game Ontology Project (GOP). While there is some overlap between the typology presented here and the GOP, I expand on some of their concepts and introduce new ones. Additionally, the typology has an exclusive focus on time, while the GOP has a more wide-ranging scope. These factors led me to the creation of an expanded and reorganized framework for the analysis of time in video games. The corresponding GOP papers will be duly cited in every case. More information can be found on the project's wiki (Game Ontology Wiki 2015).

A few terms (for example, *events* and *stages*) are borrowed from Chris Crawford's basic components of dramatic universes (Crawford 2003, pp. 264-266). However, the present typology is aimed at describing the temporality of a broad scope of video games, while Crawford's concepts are used specifically in the context of his Erasmatron storytelling engine. Thus, the terms used here differ in meaning and function.

CHANGE OF STATE

This first category, *change of state*, is the necessary foundation for all others. Following Jesper Juul, I have argued in the previous chapter that video games are state machines that possess different moving parts that interact with each other, some controlled by player input and some by the game itself. These moving parts are routines written in the game's code and represented in its audiovisual layer. The capacity of the screen to change state according to what the code dictates allows for a fast and intuitive representation of the changes coded in abstract machine language. As a result, zeroes and ones transform into colorful fictional worlds with running and jumping characters, governed by physical laws much like those in the real world.

Events

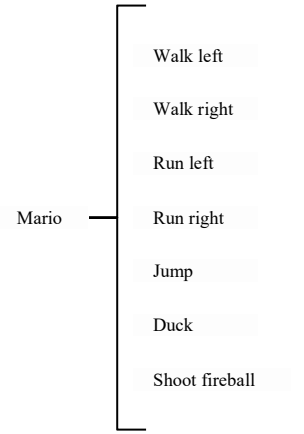
Events are the basic building blocks of a game's temporality. They result from the change of state of an entity in the game and are usually identified with an action, such as walk, run, jump, shoot, or talk.

Entities can act and react in numerous ways. The actions of one entity may elicit a reaction from another (hitting a box might break it), producing a causal chain of events that moves in one direction (character hits box with crowbar, box breaks). Some entities can answer back when acted upon (character one hits character two with crowbar, character two loses hit points and hits character one back), which in turn can produce another reaction in the first part (man two hits back, man one hits back again, and so forth until one falls). The latter is a chain of events that moves in two directions—that is, an interaction in the second sense mentioned in the introduction.

One or more of the entities in the gamespace are usually controlled by player input, while many others are controlled directly by the program itself. It is up to the designer to determine what events the player can prompt with the controller and how the game world will react to these actions.

A list of the events that a player can start in SUPER MARIO BROS., for instance, would look something like in figure 1.6.

Figure 1.6: Events players can start in SUPER MARIO BROS.

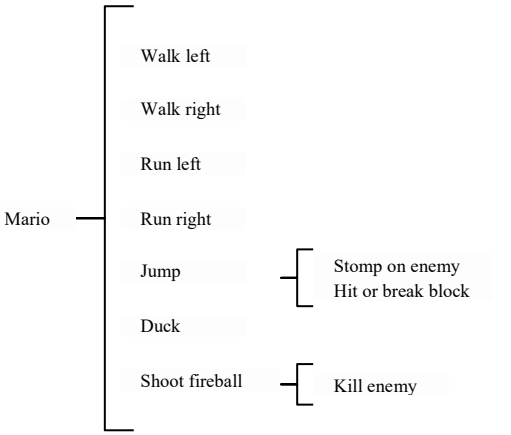


It is a basic skill set, but some events enable the player to perform others that are context-dependent. Jumping, for instance, allows the player to hit blocks above the player character or land on enemies. By hitting brick blocks, the player can break them or extract coins from them, for example. Landing on enemies typically eliminates or incapacitates them, depending on the enemy type.

Therefore, some events are a *direct* effect of the player pressing a button, and others an *indirect* effect, because they require for a direct action to influence a player-independent entity.[2] One could then say that the player can kill enemies in SUPER MARIO BROS., but only by jumping or shooting a fireball first. Jumping and shooting fireballs are *direct events*, and eliminating an opponent is the *indirect event* that can take place as a result (figure 1.7).

2 Jesse Schell distinguishes between "operative" and "resultant actions" (2008, pp. 140-142). These terms are roughly equivalent to the direct and indirect events as defined in this typology.

Figure 1.7: Direct and indirect events in SUPER MARIO BROS.



Pace

Events in video games can take place at different paces. Some games, such as QUAKE (id Software 1996), might require quick reflexes from their players, and others can be comparatively slow-paced and more concerned with strategic thinking, as is SIMCITY (Maxis 2013). Moreover, games themselves can exhibit internal changes of pace. Therefore, pace in a video game can either be *constant* or *variable.*

QUAKE has a *constant pace* because events in the gameworld always occur at the same speed. Conversely, SIMCITY has a *variable pace*, since the player can choose between different rates at which events can unfold.

Game scholars José P. Zagal and Michael Mateas refer to this feature as a part of their *manipulating gameworld time* category in their analysis of gameplay time (2010, p. 856-858). In it, they mention games like MAX PAYNE, where the player can slow down time in order to perform spectacular stunts, such as dodging bullets and aiming and shooting while in the air after performing a jump (this subject will be further analyzed in section 3.1, with relation to time perception in dangerous situations).

Figure 1.8: SIMCITY.

Source: https://www.gameaxis.com/reviews/review-simcity-2013-win/

A city in 2013's SIMCITY. The buttons that control the pace of the game are on the bottom-left corner of the screen.

Cycles

Events in gamespaces are often cyclical. Non-interactive background objects, for example, tend to have looping animations—an effective and resourceful way of displaying constant motion.

Some cyclical events are more than just window dressing and actually play a role in the game mechanics. Non-player characters can move in cycles if they are patrolling an area, for instance. As Mark J.P. Wolf points out, "learning the patterns of behavior and working around them is usually part of the game" (Wolf 2002b, p. 81) Additionally, "[c]ycled action builds player expectation and anticipation" (ibid.).

The most prominent example is perhaps the day/night cycle, which is most common in open-world games. Many of these games feature character skills that only work at night or during the day and, thus, some events can only occur in particular time windows. Day/night cycles are also reliable guides to tell the passage of time in a game, since they are regular occurrences—just like in the real world. For this reason, games with day/night cycles tend to use days (or even weeks) as units of measurement for some events.

Figure 1.9: THE WITCHER 3: WILD HUNT – BLOOD AND WINE.[3]

THE WITCHER 3: WILD HUNT and its two expansions feature day/night cycles that can influence aspects of the gameplay. This screenshot was taken at 10:37 am (in-game time).

BLOOD AND WINE (CD Projekt Red 2016), the second expansion for THE WITCHER 3: WILD HUNT (CD Projekt Red 2015), uses the hours of the day, as well as days and weeks, as a measure of the passage of time. The quest called "Paperchase" requires that the player waits one week in-game before continuing. On a different quest, the Witcher contract "Big Game Hunter," the player is invited to a picnic that will take place the following day at noon in order to celebrate the successful mission. Nevertheless, the player can choose to attend the celebration any day, as long as it is noon, without fear of missing it. Thus, the important datum in this quest is "at noon," a temporal coordinate that repeats itself once every day/night cycle.

3 Unless specified otherwise, game screenshots are of my authorship.

Some multiplayer games with persistent worlds do have events that the player can miss. Zagal and Mateas (2010, p. 852) illustrate this with ANIMAL CROSSING (2002) for the GameCube console:

"ANIMAL CROSSING contextualizes its fictive time with respect to that of the real world. When the player first starts the game, he or she must enter the current real-world date and time. From that moment, the gameworld tracks time just as a clock in the real world would. The synchronization is such that by not playing on December 25, the player misses all the Christmas day in-game activities."

Christmas events, such as the apparition of Jingle the reindeer (Animal Crossing Wiki 2017), could be missed if the player did not enter the game on December 25. One whole year would have to pass until the repetition of the event.

Pause

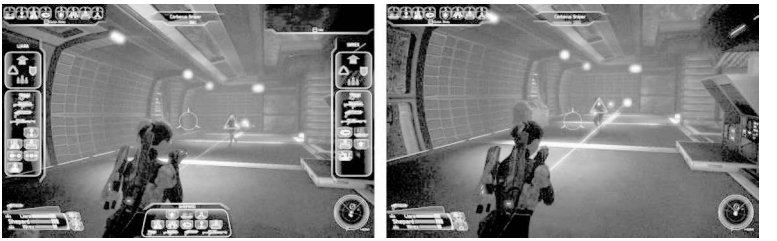
Just as the passage of time can be sped up and slowed down in video games, it can also be paused. Typically, the pause function is used in single player games as a tool to take a bathroom break or grab a snack, since it precludes events from occurring while the player is away.[4] *Pause* is part of Zagal and Mateas' *manipulating gameworld time* category: "Temporarily suspending, or pausing, gameworld time (relative to real-world time) is another common temporal manipulation. The idea is that no events occur in the gameworld while gameworld time is paused" (2010, pp. 856-859).

Some games, however, incorporate pauses as part of their gameplay. DISHONORED (Arkane Studios 2012) and its sequel, DISHONORED 2 (Arkane Studios 2016), put the player in control of characters with special powers, such as teleportation and the ability to see enemies through walls. Corvo Attano—the main character in the first game and one of two available characters in the second—has the capacity to stop time around him. This power freezes everything in place while allowing Corvo to walk around enemies as if they were statues—and dispose of them if the player wishes. But this is more of a partial than a full pause, since the game is still running and the player can navigate the environment. Additionally, the player needs to act swiftly before enemies resume their motion, since the gameworld remains static only for a limited period of time.

4 Online multiplayer games lack a pause function, since the gameworlds are populated by other players. Especially in cooperative games, where players act in groups, the phrase "going AFK" (away from keyboard) is commonly used to announce a break.

MASS EFFECT allows the player to bring the action to a complete halt in order to strategize during combat sequences (figure 1.10). By pressing and holding a button, players open a menu that allows them to select skills and weapons, and issue commands to teammates. While this menu is open, the fictional gameworld is completely paused. This brings me to the next element in this category.

Figure 1.10: MASS EFFECT.



Left: The menu that allows players to select skills and weapons, and command squadmates. While this menu is open, the action in the three-dimensional gameworld is paused. Right: An instant later, after the menu is closed and the action resumes.

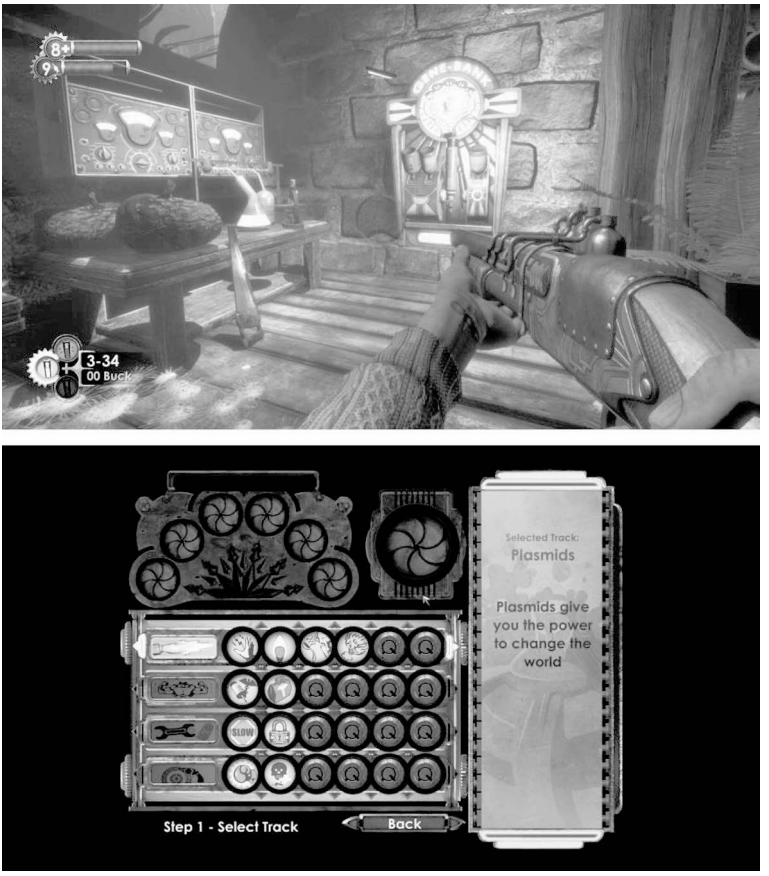
Layers

Interaction with games can happen through overlapping layers, which can have completely different interfaces and events, and unfold at diverse paces. Layers can, for example, be menus that add another aspect to the interaction with the gameworld, such as inventory administration, or the selection of dialogue options.[5] In the above-mentioned MASS EFFECT, while players interact with the menu layer, the layer of the three-dimensional gameworld remains paused, giving them time to make complex decision reactions—to use Pöppel's terminology. Once the layer is closed, action resumes at a normal pace, better suited to simple reactions—though decision reactions still take place.

5 These menus are part of what Jesse Schell calls the virtual interface: "a conceptual layer that exists between the physical input/output and the game world" (2008, p. 224). That is, the physical interface would be constituted by the keyboard and mouse, or the game controller, and the virtual interface by the menus, buttons, and information displays of the game that are not a part of its fictional world.

In BIOSHOCK (2K Boston 2007), for instance, the main layer is the first-person camera view of the three-dimensional world. But the game also has a number of two-dimensional layers (figure 1.11) in the form of menus used to, for instance, upgrade weapons and skills. Additionally, there are menus to select which skills are active, since the player can only have a limited number of active powers at any given time. The decisions made on these menus change the game state and influence which direct events the player character can initiate in the first-person layer.

Figure 1.11: BIOSHOCK REMASTERED (2K Boston 2016).



Top: First-person view. This is the main layer of the game. Bottom: The menu layer used to activate skills.

Events that occur on one layer do not necessarily affect other layers, but layers do tend to influence each other. Other examples of layered interfaces are hacking minigames and character screens that let the player choose which attributes of their character to improve (for an example of a skill tree, see figure 1.20).

Reset

Unlike time in the real world, which runs strictly in one direction, gametime is reversible. All of the events that took place in a game can be undone by starting the game anew, or by saving the game state at particular points in time and then loading them. Players typically load saved states whenever an event occurs that leads to an undesired outcome or impedes further play. To this purpose, numerous games feature "load game" menus or "quick load" keys that load the last saved game state when selected. This action is typically not a part of the fictional world of the video game.

In the early days of gaming, before save states were a common thing, players relied on *lives* and *continues* to keep their progress. In these cases, games allowed players to restart a stage or a stage section at the cost of one of these resources. SUPER MARIO BROS. gives the player three lives at the start of the game. Additional ones can be acquired by collecting green mushrooms or one hundred coins. Each time Mario dies, he loses a life and returns to the beginning of the level. SONIC THE HEDGEHOG (Sonic Team 1991) has checkpoints marked by posts placed in each stage. If Sonic loses a life after activating one of the checkpoints, he does not return to the beginning but spawns at the post. Besides lives, SONIC THE HEDGEHOG features continues, which are also available in limited quantities. After losing all lives, the player can use a continue and start the level from the beginning. Checkpoints do not work in this case. SONIC THE HEDGEHOG is a console game, but continues are also common in arcade games, where they cost players money. By paying again, players are granted a new set of lives and the possibility to maintain their progress (compare Atkins 2007, p. 246).

Some games incorporate the reset function as an in-game event that can be directly performed by the player-character. PRINCE OF PERSIA: THE SANDS OF TIME (Ubisoft Montreal 2003), BRAID (Number None, Inc. 2008), and LIFE IS STRANGE (Dontnod Entertainment 2015), for example, all have characters with the capacity to rewind time. As a result, the reset function becomes a diegetic event that the player can initiate with the press of a single button. The resulting effect is similar to the rewinding of a VHS tape, with past events unfolding again in the opposite direction.

Chapter 2, section 2.2 will expand the reset concept and analyze some of the consequences that it can have for gameplay and storytelling.

SPACE-TIME

Space and time are interconnected in our minds. Psychologist Steven Pinker remarks that this relation manifests in spatial representations of time, such as calendars and hourglasses (Pinker 2007, p. 191). Language also offers a window into the mental connection of space and time in the form of spatial metaphors like "a long/short time," "time goes by fast/slowly," or "time flies." "A long time" is a metaphor of time as distance and "plenty of time" describes time as an amount. A particular point in time can be ahead of us, behind us, or already here. All of these metaphors share a direct relation to space (see Clark 1973, pp. 48-50).

We can also understand time as something moving through us, or as something within which we are moving. When we interpret time as an entity moving past us, we are using the *time-moving metaphor,* which manifests in expressions like "time flew by" or "noon crept upon us" (Hasplemath 1997, p. 59; Lakoff and Johnson 1980, pp. 41-45). If we see time as something through which we move, then the *ego-moving metaphor* is implemented. In such cases, we can say that "we are approaching the end of the year" or "he is going through a rough time" (ibid.). Zagal and Mateas note that research by psychologists Boroditsky, Ramscar, and Frank (2002) has shown that spatial experiences can influence the metaphors that people use. A person moving on an office chair through space is more likely to think of time in terms of the ego-moving metaphor. Someone pulling a chair towards them will tend to think in terms of the time-moving metaphor. "Thus," Zagal and Mateas infer, "the player's experience of time can potentially be manipulated or influenced through game design via tasks that trigger specific forms of metaphoric temporal cognition" (Zagal and Mateas 2010, p. 848).

The *tau* and *kappa* effects provide further evidence of the connection of time and space. To produce the tau effect, three flashes of light are shown that are at equal spatial intervals but different temporal intervals. Between the first and second flashes, the temporal interval is shorter than between the second and third. This creates the illusion that the first and second flashes are placed at a shorter distance than the second and third (Helson 1931). The kappa effect occurs when the time intervals between stimuli are constant, but the distances between them vary. The temporal interval seems longer between the flashes that are located

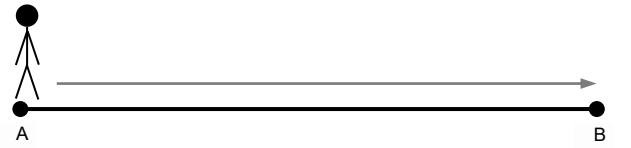
further apart, while the inverse is true for flashes that are closer together (Cohen 1955).

Henry Jenkins stated that "[g]ame designers don't simply tell stories; they design worlds and sculpt spaces" (Jenkins 2004, p. 121). Through the sculpting of these spaces, they also determine the temporal sequence of events in their games and influence the player's experience of time. The category of *space-time* analyzes features of video games that pertain to this connection.

Navigation

Navigation is one of the most common actions that players can perform in video games. Given a gamespace (whether two or three-dimensional), any object placed in it at any particular point can navigate by altering its coordinates. Thus, if a character is located at a point A, as seen in figure 1.12, by pressing right on the controller the player can move the character to a point B. The relation between navigation and time lies in the simple fact that, all other things being equal, the farther away point B is from starting point A, the more time it takes to arrive at B.

Figure 1.12: Navigation in a two-dimensional game.



Games that feature a form of navigation resembling the one shown in figure 1.12 are *unidirectional*, since the movement of the player character tends to trace a line in one direction. Pure unidirectional navigation is very rare. One example is the game LINE WOBBLER, which is described by its designer as "a one-dimensional dungeon crawler game with a unique wobble controller made out of a door-stopper spring and a several meters long ultrabright LED strip display" (Robin Baumgarten's Game Experiments n.d.).[6] Platformers like SUPER MARIO BROS. also come close to unidirectional navigation. Even though it is a two-dimensional game in which players can jump, and the levels have platforms placed at different altitudes, the direction of movement takes place predominant-

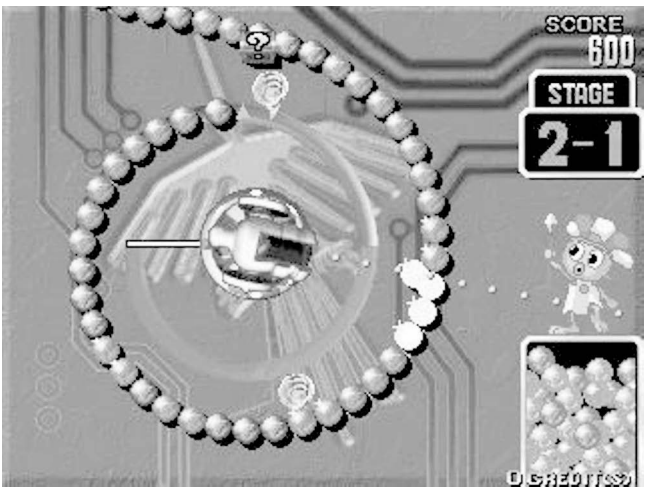
6 LINE WOBBLER can be seen here: http://aipanic.com/projects/wobbler (accessed March 20, 2018).

ly from left to right. The primary objective in the game is to reach the end of each level, which is consistently placed at the furthermost point on the right. This is accentuated by the fact that backtracking in this game is limited by a camera that only pans to the right, blocking access to portions of the level that lie to the left of the frame.

Games feature *multidirectional navigation* when they don't prescribe a direction of movement, but allow the player to choose from one of a variety of available directions. These games might still have an endpoint at a predefined location in the gameworld, but it is largely up to the player to choose how to traverse the world. An early example of multidirectional navigation is PAC-MAN. Games can also feature both types of navigation simultaneously. In chess, for instance, pawns move only forward (normally in a straight line, but diagonally when capturing), whereas the queen can move in eight different directions at any time.

Hence, navigation requires a space and an entity with the capacity to change position in it. Sometimes the space is displayed in its entirety on the screen, such as in SPACE INVADERS or PONG (so-called single-screen games), and the player characters navigate within the frame. In other games, the camera moves together with the player character, like in first and third-person perspective games such as TOMB RAIDER (Core Design 1996) or DOOM (id Software 1993). Finally, the camera and the player-controlled character(s) can also navigate independently, as in real-time strategy games like COMMAND & CONQUER (Westwood 1995).

Figure 1.13: PUZZ LOOP.



Source: https://www.arcade-museum.com/game_detail.php?game_id=9165 (accessed December 1, 2017).

Distinguishing the narrower term *navigation* from the broader *movement* is key. Navigation is a form of movement, but there can be movement without navigation. One example of a video game that features a player character which moves but does not navigate is PUZZ LOOP (Mitchell Corporation, 1998). In it, the player controls a rotating cannon that shoots marbles of different colors (see figure 1.13). The objective of the game is to shoot at a spiraling line of marbles that moves from the edge of the screen towards the center, where the cannon is located. If the line reaches a particular point near the center, the player loses. To prevent this, the marbles can be destroyed if the player hits two or more consecutive ones of the same color with one that shares that color. In this game, the player can move the avatar solely by rotating it, and the camera is static, so navigation with the player character is not possible. Nonetheless, navigation is common in video games, and it has been featured as a central gameplay mechanic by countless titles from very early on.

The seminal video game SPACEWAR! (Russell 1962), for instance, featured spaceships that could navigate a two-dimensional space while firing rockets at each other. Navigation in SPACEWAR! is *continuous* because the space in the game is not segmented and can be traversed in its entirety. To navigate from A to B, the spaceship needs to fly the distance.[7]

Other games feature *discrete navigation*, which means that space is divided into smaller units and the player character moves in increments of one or more of these units. Discrete navigation can thus be counted in amounts of units—for example, the number of squares. This is typical of board games such as Chess or Checkers, both in their analog and digital versions. MYST (1993) is an example of a video game with discrete navigation that was not previously an analog game. In MYST, players navigate the gameworld from a first-person perspective. However, players do not control the character directly, but a mouse cursor instead. By clicking on the image in the direction they want to go, the view of the gameworld is replaced by a new frame that shows the player character's new location or orientation (figure 1.14). Thus, movement in MYST can be tallied in discrete units that could be called *screens* or *frames* (see Wolf 2002b, p. 80).

7 SPACEWAR! also features a hyperspace mechanic that teleports the ship to a random point on the screen. Given the randomness of this feature, it is more useful for dodging rockets than for navigation.

Figure 1.14: MYST: MASTERPIECE EDITION (Cyan Worlds 2009).

This progression of four consecutive screens shows navigation from a courtyard with a fountain into a library. The sequence starts at the top-left and ends on the bottom-right.

Since their early days, game spaces have gradually grown in scale. Some game worlds are now so large that moving from A to B can take upwards of an hour. YouTuber TheyCallMeConnor (2015) uploaded a time-lapse video walking through the most extensive continuous map in THE WITCHER 3: WILD HUNT, from one extreme to the other. He reports having walked for 45 minutes and 30 seconds.[8] Games of this size commonly implement fast travel mechanics—a gamic version of the filmic jump cut that is activated by players whenever they want to jump to another location in space. These mechanics replace continuous spatial navigation with discrete spatial navigation.

In THE WITCHER 3, for example, players need to interact with street signs, which open the map and allow them to jump to any other activated sign. These signs are activated when Geralt, the protagonist, is close to them, so the player needs to travel to each signpost at least once without fast travel.

GRAND THEFT AUTO V (Rockstar 2013) offers a form of fast travel grounded on another form of transportation. If players don't feel like driving, walking, or flying somewhere, they can always call a cab, which costs a small amount of in-

8 In the end of the video, TheyCallMeConnor lists other game maps he has walked through, such as GTA IV's Liberty City (one hour and sixteen minutes), or Fallout New Vegas (one hour and nine minutes).

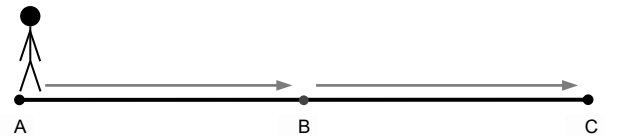
game currency. Taxis can take the player character in real time to its destination, in which case the player can vicariously observe how the car moves through the city. But they also offer the option to spend an additional amount of money to skip the drive and jump to the destination.

Fast travel mechanics usually advance time in the game. Both THE WITCHER 3 and GTA V feature day/night cycles, which move forward in accordance with the covered distance whenever fast travel is used. That is, fast travel is in these cases not only a means to jump in space, but also in time.

Location

With the principle of navigation now established, it follows that the further away an entity is placed from the player character the later it will be encountered. Going back to the scheme introduced above, if the player character starts at a point A in space and walks in a straight line, it will arrive earlier to a point B placed closer to the starting point than to a point C at a greater distance (figure 1.15). By locating objects and characters at different relative distances from the starting point of the game, designers can determine the sequence in which the player will come across them.

Figure 1.15: The principle of location.



All else being equal, the player will reach a point B earlier than a point C if point B is located closer in space to the starting point A.

Video games typically add difficulty to this formula, tying the principle of location to the notion of *progression* that I will describe later under the *conditions* category. The further away a challenge is, the more difficult it will tend to be: Platforms in platformers get smaller and further apart, enemies in *beat 'em ups* become stronger or appear in higher numbers. While a challenging section in a game might be followed by a rather easy one, the difficulty of a game tends to increase as the player progresses.

The items that the player character obtains have a similar transitive relation to space that benefits the player: The further away a weapon or item is, the more powerful it will tend to be. Traditionally, the player character starts a game with a single, relatively ineffective weapon or skill and acquires better ones as the player makes progress. Weapons obtained later in the game tend to do increasingly more damage, which in turn compensates for the increasingly challenging enemies. This characteristic of games will be discussed further in chapter 3.3, Chekhov's BFG.

This type of structure in which the progression of events and difficulty are tied to location in space is the backbone of a vast number of video games, and it is the feature that defines linear games. Think for example of HALF-LIFE (Valve 1998), SONIC THE HEDGEHOG, CRASH BANDICOOT (Naughty Dog 1996), MEDAL OF HONOR: ALLIED ASSAULT (2015, Inc. 2002), or PRINCE OF PERSIA: THE SANDS OF TIME. In all of these games the player's objective is to go from A to B (regardless of the plot) and events will not occur until the player reaches the location in space where they are to take place. Additionally, the further away the player strays from point A, the harder the game becomes.

Figure 1.16: PORTAL.

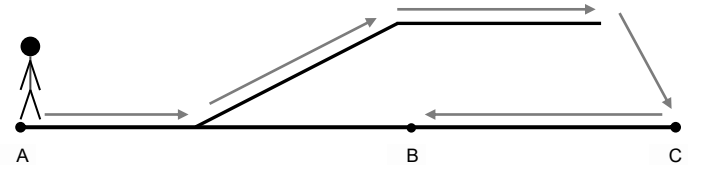
The video game PORTAL (Valve 2007) adds a fascinating ingredient to navigation and the principle of location by allowing players to open gateways that connect remote sections of space. As seen in figure 1.16, the player can walk into the blue portal and come out through the orange one in the background. In this particular puzzle, the player needs to press two red buttons (one of them to the right of the image) within a short time window. But that interval is not long enough to navigate from one button to the other, which forces the player to use

portals to shorten the distance between both points, consequently reducing the time it takes to navigate from one to the other.

Space-Time Obstacles

Even in games with unidirectional navigation, space is rarely an empty canvas that allows the player to move in a direct line. Space in games presents obstacles that need to be surmounted for the player character to continue pushing forward. These space-time obstacles make navigation varied and challenging. The fun part of a game is what happens between the starting and the ending points: Shooting monsters, moving from platform to platform, fist fighting, interacting with other players or non-player characters, or simply walking peacefully and enjoying the sights and sounds. Space-time obstacles often break the simple location principle, rendering points in space that are close to the avatar harder to access. Thus, as seen in figure 1.17, a point B close to the starting point A can be accessed later than a point C that is located further away.

Figure 1.17: Location and a space-time obstacle.



A simple change in the design of a level can alter the sequence in which points B and C are encountered. Point B is still closer in space than point C, but further away in time because of the space-time obstacle. From the perspective of the designer, the design of the game space needs to agree with the actions that the player character can perform. In a platformer like SUPER MARIO BROS., for example, where running and jumping are core mechanics, levels consist mostly of pits to jump over and platforms to jump onto. Accordingly, the actions that the player character can perform will shape how players perceive the space. In a game with no jumping mechanics like MASS EFFECT, even a low wall can be a space-time obstacle, so designers use them for different purposes than in SUPER MARIO BROS. Pits and walls in MASS EFFECT are insurmountable barriers that prompt players to look for an alternative way to their destination. In SUPER MARIO BROS. hindrances like these will generally prompt players to try to jump over them.

Often, games let players know that their goal is closer to them in space than it is in time—when an objective is placed behind a locked door that is close to the start of a level and the player needs to search for the key first; or if powerful enemies block access to a particular location to prevent the player from navigating it without advanced skills or weapons. These strategies (explored further under *progression* in the *conditions* category) can be used to generate anticipation and add intrigue to the gameplay experience.

The so-called *metroidvania* games are characterized by this type of design. Metroidvania is a portmanteau of METROID and CASTLEVANIA,[9] the two franchises that helped popularize the genre. These games are characterized by gameworlds with multidirectional navigation and obstacles that require specific skills or weapons to be surmounted. In METROID (Nintendo, 1986), for example, some doors can only be opened by shooting them with a missile. Since the player character is not equipped with a missile launcher in the beginning, players need to explore the gamespace in search for the weapon in order to open these doors.

Triggers

Triggers are entities in the gamespace that respond to the presence of the player character (or other entities) by initiating an event.[10] They can be *visible* or *invisible*.

Visible triggers are represented for instance by buttons, levers, or pressure pads that need to be activated voluntarily by players. In the case of a pressure pad, this is achieved by placing an entity (such as the player character) on top of it; buttons and levers usually require the avatar to be in their area of influence and the player to press a keyboard key or button on the controller. Since they are visible, the player can decide when to activate them and, if they are not an

9 In the case of CASTLEVANIA the term metroidvania refers in particular to CASTLEVANIA: SYMPHONY OF THE NIGHT (Konami Computer Entertainment Tokyo 1997), the first game in the series to feature the multilinear structure characteristic of metroidvania games.

10 The term *trigger* is borrowed from game-engine vernacular, but it is used here in a more general way that eschews specific technical details. The Unreal Engine 4 Documentation, for example, defines them as follows: "Triggers are Actors that are used to cause an event to occur when they are interacted with by some other object in the level. In other words, they are used to trigger events in response to some other action in the level." (Unreal Engine 4 Documentation 2014-2017). The Valve Developer Community Wiki defines triggers as "entities which respond to the presence of other entities." (Valve Developer Community Wiki 2016).

integral part of a primary mission or objective, they can also be ignored altogether. Triggers such as these can be used to open doors, call elevators, start machines, or turn lights on and off.

Invisible triggers are areas of influence that react to the player character's presence. They are typically brushes—that is, primitive shapes like cubes, cones, or spheres—that are invisible and intangible, and can thus be placed in levels to sense the player character's presence surreptitiously. Since they do not alert players of their existence in any way, they are used to initiate scripted events that are meant to give life to the gameworld—like a narrative booby trap.

Scripted events are events that are previously orchestrated by the developer and occur whenever a trigger senses the player character's presence. In that way, the gameworld comes to life and events that do not involve the player directly can unfold as in-game action.

The first-person shooter HALF-LIFE was acclaimed for its groundbreaking use of scripted events, among other things, back in 1998.[11] In an early sequence of the game, the player needs to escape a research facility after initiating a multidimensional debacle during an experiment. The corridors of the facility are brought to life by myriad scripted events, such as exploding machinery, falling elevators filled with passengers, and security guards shooting scientists turned into zombies by parasitic creatures called *headcrabs*. All of these events wait for the player to arrive before they start happening. They are initiated by both visible and invisible triggers placed in their vicinity, ensuring that the player does not miss a single moment of the chaotic spectacle.

Following the principle of location, triggers orchestrate the spatiotemporal sequence of events in a game. They also make gametime (especially in single-player games) fundamentally player-centric. I will return to this last point in the next section (1.3).

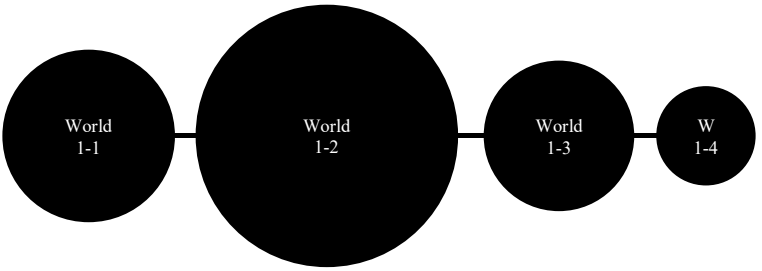
11 Roy Dulin's review of HALF-LIFE for Gamespot states that "[t]here are scripted events in the game. There are opening and closing scenes. But they all occur naturally within the game environment. It may sound simple, but it goes a long way toward helping create a believable world." He later adds that scripted events "bolster the illusion of reality." (Dulin 1998). IGN's review cheers: "The sheer number of hand-scripted events and little scenes keeps the action moving, giving you a reason to keep playing, if only to see what could possibly happen next. I haven't had so much fun playing a game in years. I have not been frightened by a game in years. I have not dreaded corners like I have dreaded corners in this game in years. HALF-LIFE is a superbly ambient game" (IGN 1998).

Stages

Space in video games is often partitioned into disconnected stages. Game scholars Zagal, Fernández-Vara, and Mateas, have dubbed this aspect "spatial segmentation" (Zagal et al. 2008, pp. 181-186). In the sense used here, stages are enclosed gamespaces where events unfold.[12]

Since player characters can only be in one stage at a time, stages are played sequentially. But the order is not always predefined by the game designers. Stages in SUPER MARIO BROS., for example, follow a strictly *linear progression,* which typically signals a gradual difficulty increase from stage to stage in games of skill.[13] Other games, like MEGAMAN (Capcom 1987), allow players to choose stages from a menu and play them in any order. This results in a *multilinear stage progression*—that is, a stage progression that players can structure in several different ways. The final result is naturally always a linear sequence of stages, albeit one configured by player choice.

Figure 1.18: Stage progression in SUPER MARIO BROS.' first world.

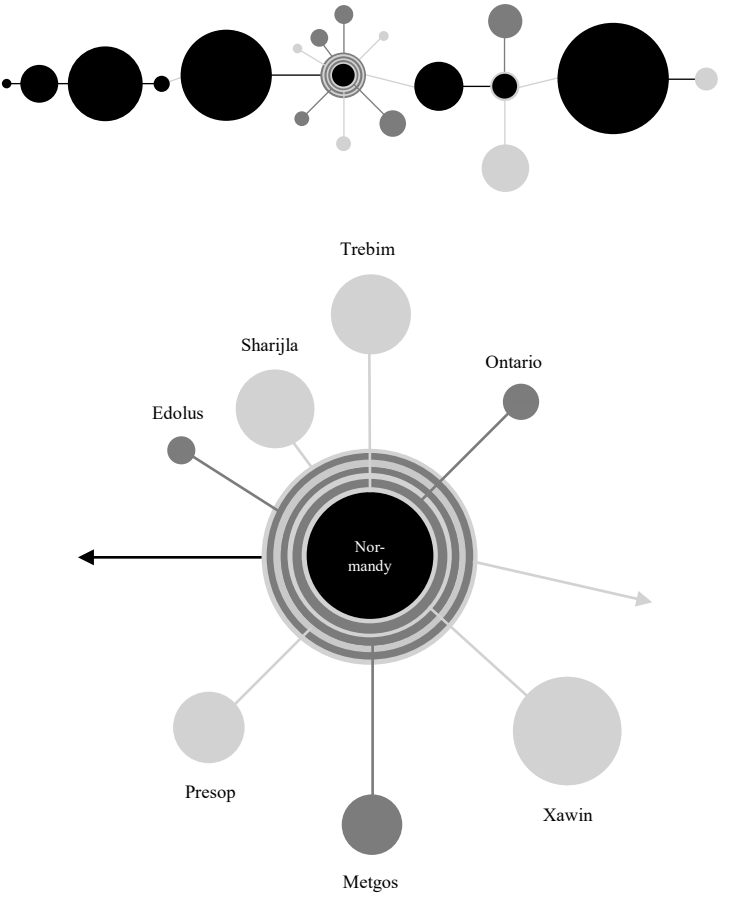


12 Games sometimes label the different spatial segments as *stages*, but several other terms such as *level*, *world*, *zone*, *area*, or *map* have been used with the same meaning. The word "level" is perhaps the most commonly associated with spatial segmentation, but this term can be ambiguous, since it is also used to refer to the levels of a player character (as in "my SKYRIM character is level 15") or to degrees of difficulty, as Zagal, Fernández-Vara, and Mateas have remarked (2008, p. 183). I have chosen the term "stage" for its meanings as both a space where action unfolds and a step in a progression.

13 It should be noted that stages can be skipped in SUPER MARIO BROS. by using secret warp zones. However, the different stages are numbered from 1-1 to 8-4 and can only be skipped in ascendant order. Once on stage 4-1, for example, it is not possible to return to stage 3-4, and the game always starts in stage 1-1 and finishes in stage 8-4.

Figures 1.18 and 1.19 illustrate the stage progression in SUPER MARIO BROS. (*linear progression*) and MASS EFFECT (*multilinear progression)* with data extracted from my personal playthroughs. Stages are depicted as circles whose diameter is proportional to the time it took me to finish them. The progression in SUPER MARIO BROS.' first world, which is made up of four stages, is shown in figure 1.18.

Figure 1.19: Stage progression in MASS EFFECT.



Top: The first ten hours of MASS EFFECT. Bottom: Detail of the radial configuration of secondary missions around the Normandy stage.

Figure 1.19 (top) shows the first ten hours of MASS EFFECT (2007). A few new elements have been added to the representation system in this case: 1) The lines connecting different stages are horizontal when the progression is linear, and diagonal when I could choose between two or more possible stages to play next. 2) Black circles represent main (mandatory) missions, while grey circles represent secondary (optional) missions. 3) MASS EFFECT is characterized by having a recurrent stage, the spaceship Normandy, which functions as a center of operations where players develop relationships with their crew, upgrade armor and weapons, and choose which stage to complete next. The graph represents this peculiarity with the radial configuration seen in the middle of the figure.

Aside from the Normandy, stages in MASS EFFECT are celestial bodies or space stations scattered across the galaxy, to which the player can travel with the spaceship. After completing the main missions in each stage, the player returns to the Normandy. As long as the player visits stages that only include secondary missions and returns to the Normandy, the main story arch does not advance. In figure 1.19 (bottom), the first secondary mission that I chose (Trebim) is located at 12 o'clock, and the progression moves in a clockwise fashion. The circle in the middle represents the Normandy, and its layers in different values of gray show each time I came back to the ship after a secondary mission. Once I chose a new stage with a primary mission, the timeline progresses to the right.

CONDITIONS

So far, the two previous categories have analyzed how video games can change state on their mediated layer and thus represent events—which can be influenced in different ways—and how those events can be structured in time via their organization in space. This third and final category looks at how video games encourage players to pursue specific outcomes and avoid others by setting *conditions*. These conditions constitute the final layer that structures the temporality of video games.

Time Gauges

Time gauges are elements that change state in a regular way and inform the player about the amount of time left within a limited window. These can take the form of clocks, progress bars, or day/night cycles. Time gauges are commonly used to set conditions on gameplay. When a timer runs backward, it informs the player that something will happen when the display reaches zero (see Wolf

2002b, pp. 88-91). SUPER MARIO BROS. is a classic example of the use of countdowns. In this game, players have limited time to navigate through each stage. If the end of the stage is not reached by the time the countdown ends, the player loses a life. The game underscores the final moments of this countdown by doubling the music's tempo, which warns players of Mario's imminent death and increases tension.

The countdown is also featured in escape sequences in which a self-destruct timer of a facility initiates, prompting the player to rush to the exit to avoid being buried alive in the rubble. The Metroid games often feature such escape sequences, one example being the final moments of SUPER METROID. After the player defeats the final boss, Mother Brain, a three-minute countdown starts together with the sounding of an alarm, screen shaking, and explosions. To finish the game, the player needs to reach the exit before the time expires. Sports games like FIFA 17 (EA Canada 2016) structure matches in two half times—just like real-life football but with much shorter default spans. Thus, players have limited time to complete the objective of scoring more goals than the opposing team (Zagal et al. 2008, p. 180). In the game DEAD RISING (Capcom 2006), players need to finish the main campaign in 72 in-game hours, or they will fail.

Another typical use of time gauges is the oxygen bar. Video games with swimming mechanics that allow the player character to dive under water usually feature an indicator that signals the player when the player character is starting to run out of oxygen. These time gauges can take different forms, the most common perhaps being a depleting blue progress bar, as seen in THE WITCHER 3.

Zagal, Fernández-Vara, and Mateas have rightly pointed out that, in cases when there is a time limit to certain events, time acts as a resource, just like ammunition or hit points (2008, pp. 180-181). But time gauges do not need to be countdowns to influence players. Some games reward those players who finish a stage below a particular time with a good grade or with extra points, encouraging players to be fast and efficient. An example is DMC: DEVIL MAY CRY (Ninja Theory 2013), and time trials in racing games such as FORZA MOTORSPORT 7 (Turn 10 Studios 2017).

The above-mentioned day/night cycles are also an example of a time gauge that can have a direct influence on gameplay. Zagal and Mateas (2010, p. 851) exemplify this with the game KNIGHT LORE (Ultimate Play the Game 1984), in which the player character is human during the day and turns into a werewolf at night. As a werewolf, the player character is attacked by non-player characters that would otherwise remain peaceful (Parrish 2007). The player also has a maximum of forty in-game days to finish the game, which entails finding a wizard that can rid the protagonist of his lycantrophy (Computer & Video Games 1985).

Animations can also work as time gauges, a typical case being the reload animation in shooters (see Zagal and Mateas 2010, p. 853). In HALF-LIFE, for example, once the clip of the handgun is depleted, the player needs to press the key assigned to the reload function in order to continue shooting. While reloading, the player character's hands engage in an animation whereby they replace the old clip with a new one. This animation renders the player character unable to attack until the gun is loaded again.

Turns

I have so far treated real-time as the default way in which time passes in games. However, games or game sections can be structured in *turns* when the passage of time takes place in discrete increments (compare Zagal et al. 2008, p. 179). Turns could be seen as the temporal equivalent of discrete navigation.

Turn-based games stand in distinction to real-time games. This classification is most common in strategy games, which can be largely divided into real-time strategy (RTS)—e.g., COMMAND & CONQUER or AGE OF EMPIRES (Ensemble Studios 1997)—and turn-based strategy games—e.g., CIVILIZATION V (Firaxis 2010) or X-COM: APOCALYPSE (Mythos Games 1997). Role-playing games (RPGs) also lend themselves to this classification, since many RPGs (especially early ones) feature turn-based battling systems—DRAGON QUEST (Chunsoft 1986), FINAL FANTASY (Square 1987)—while others feature real-time combat—DIABLO (Blizzard North 1997), BALDUR'S GATE (BioWare 1998).

The archetypal case of turn-based mechanics is when players can only act one at a time. This feature is most common in board games like chess or MONOPOLY (Magie 1932). Since role-playing and strategy are genres that originated as tabletop games, it should come as no surprise that some of their video game counterparts have maintained turn-based systems.[14] When one player is acting, the others wait and observe. Depending on the game, turns can be over either when the active player says so, after a countdown finishes, or once the player has reached the maximum allowed number of events.

Time in turn-based games can also be structured in *rounds*. A *round* is over when all players have used their turn and the first player is up once again. This is common in card games, such as UNO (Robbins 1971). The overarching structure of the round can structure play by allowing for some events to take place before or after each round.

14 Analog predecessors of role-playing and strategy video games are for example DUNGEONS & DRAGONS (Gygax and Arneson 1974) and chess, respectively.

Other turn-based games allow players to act at the same time and instead organize which events can take place at which time. These types of turn-based games are also known as phase-based or WeGo games. In MASTER OF ORION (MicroProse 1993), for example, the player and the AI have one turn to simultaneously make decisions and give orders, followed by an execution turn, in which these orders are carried out.

Progression

Events in video games can evolve according to particular rules, and two types of progression usually work in tandem: *character progression* and *gameworld progression*. Gameworlds tend to develop under the assumption that players get better at the game the more they play. Thus, games of skill scale the level of challenge with tougher enemies, riskier hazards, or harder puzzles. In narrative games, the progression occurs at the level of the plot. But the player character can also develop, acquiring new abilities or items, and improving the ones it already possesses. In strategy games, where players command armies or administer cities rather than control an avatar, progression hinges on building and enhancing structures or units.

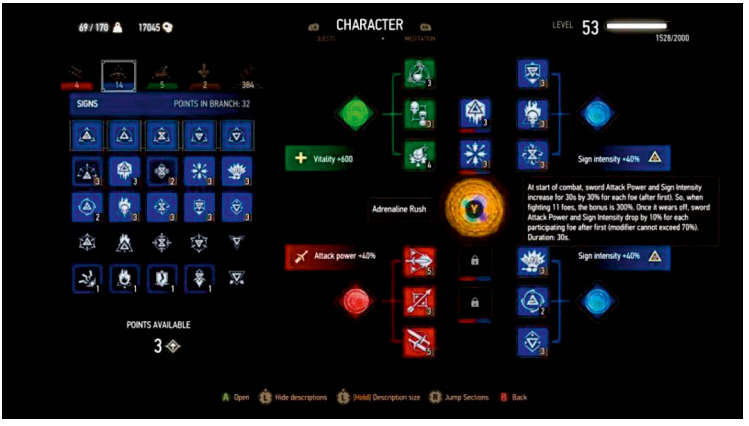
If the player character in a game can jump and acquires an ability to jump greater lengths, the jumping distances in the gameworld will likely increase as well—with higher obstacles or wider chasms, for example. In shooters, acquiring a weapon that makes more damage can be a sign that tougher enemies are coming. Video games can thus influence player expectations in two ways:

- When *gameworld progression* precedes *character progression*: If the player encounters a challenge that the player character is not yet prepared to meet, this signals that there is an item or skill upgrade to be obtained that will help them surmount the obstacle later. Games with *multidirectional* spatial design, such as those in the metroidvania genre discussed above, are structured in this way. In these games, players commonly encounter an obstacle first—such as a locked door or a chasm that is too wide to jump across—and then acquire the tool to overcome it.
- When *character progression* precedes *gameworld progression*: If the player character obtains an overpowered item or skill, the player should expect to encounter a challenge or an enemy that will put this skill or item to the test. DOOM II: HELL ON EARTH (id Software 1994), for example, never presents the player with an encounter that cannot (in theory) be overcome with the weapons that the game has already provided. The BFG9000 (Big Fucking

Gun 9000) is overpowered against most opponents and can dispose of a crowd with a single shot. Once acquired, however, it is advisable to save the BFG's ammunition for later sections that pit the player, for example, aginst a Cyberdemon, which can withstand a few shots of this weapon (just like *location*, this characteristic of games will be further analyzed in chapter 3, section 3.3, "Chekhov's BFG").

Character progression is a defining feature of RPGs, in which characters level up and obtain experience points, enabling the player to activate new skills (figure 1.20). In role playing games, player characters typically start as level one characters and can level up by completing tasks.[15] Character skills are commonly organized in skill trees, meaning that the player can chose to follow different branches, which are specialized in different ability types. In this way, the player could opt to spend experience points on a branch that develops the character's strength, one that unlocks new magical abilities, or one that improves dialogue skills. Gameworlds in RPGs are often open, so that the player can access all or a large part of it from the start. What hinders the player from going anywhere are the challenges present in each part of the gameworld. Typically, challenges in this genre are designed to be met by different skill levels. Thus, enemies in one area might be too tough for a level one character, forcing the player to level up the character before entering.

Figure 1.20: Skill tree in THE WITCHER 3: WILD HUNT – BLOOD AND WINE.



15 The term "level" in this case refers to a property of the player character.

Games can also feature the *regression* of the player character and the gameworld to an earlier state. Regression commonly happens when time is reset after failure, but it can also occur while still moving forward in the game. Some games have weapons with durability, which will break after a certain amount of use, as is the case of THE LEGEND OF ZELDA: BREATH OF THE WILD (Nintendo 2017). The fact that weapons can only be used for limited times puts players at risk of regressing to a previous state with weapons that deal less damage, lest they find a suitable replacement for their current most potent weapon.

Objectives

Video games give the player *objectives*, that is, particular game states that they need to achieve during the playthrough. Objectives can contain events that players have to actualize—for example, killing all enemies in a room or reaching the end of the stage—and events that the player needs to prevent—such as losing all hit points or running out of time. I will call the former *goals* and the latter *restrictions*. Both aspects of an objective need to be realized for said objective to be completed.

In SPACE INVADERS, for example, the objective is to prevent a descending alien block from reaching the ground. To this end, players need to shoot at and destroy the incoming extraterrestrials (goal) and avoid being hit by their lasers (restriction). Once a wave is defeated, a new one arrives. The game repeats this pattern until the player finally loses, given that there is no final alien wave.[16]

Goals in games can be varied. Some require players to collect objects, like the orbs in PAC-MAN. Other games entail defeating waves of enemies, like SPACE INVADERS. Role-playing games are characterized by the objective of upgrading the player-character to the maximum level possible. Adventure games often involve talking to non-player characters and persuading them to act as you desire, such as in THE WOLF AMONG US (Telltale Games 2013). A typical restriction is the death of the player character, which can happen in many different ways—for example, by being shot, or falling into pits or on spikes. Some games require the player character to go unnoticed by its enemies, like in METAL GEAR SOLID (Konami 1998).

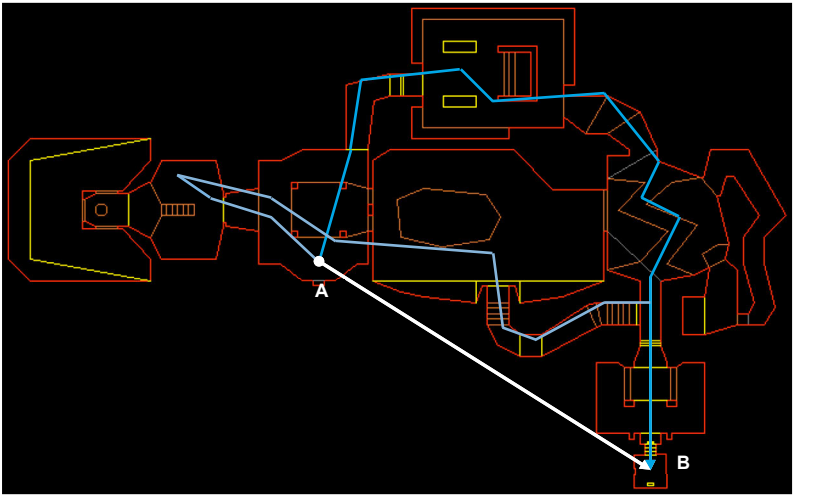
Objectives can also include a combination of several goals and restrictions. To reach the end of a stage (objective) a player might need to acquire a key (goal one) that will allow them to open a door (goal two) that is blocking the way to the exit, all while avoiding enemy attacks (restriction one) and finishing before a

16 This is popularly known as the *doomed player*, since the player is bound to lose sooner or later. The main objective is to get as far as possible in the game.

countdown is over (restriction two). In this way, objectives and goals structure gametime by determining sequences of events.[17]

In many games, going from A to B is an objective itself. When it is prescribed as a game objective, I will call the feature of going from A to B a *goal vector* (see figure 1.21). Goal vectors are displacement vectors that extend from the point where the character spawns in the level, to the point where the level ends. In DOOM, this endpoint is the location of the button that players need to press to finish the level. In SUPER MARIO BROS., the point is the iconic flagpole at the end of most stages or the ax on the far side of the bridges on which the Bowser imposters (and ultimately Bowser himself) stand in castle stages. An example of a game without a goal vector would be PAC-MAN, where stages do not end by reaching a position in space, but by eating all the orbs in the maze.

Figure 1.21: The map of THE ULTIMATE DOOM's (id Software 1995) first level.



Base image source: http://www.classicdoom.com/mapcolor/j1-pc.htm (accessed February 2, 2018).

The white arrow shows a displacement vector from the starting point (A) to the end of the stage (B). That is the goal vector. The blue lines show two possible ways through the map.

17 Zagal, Fernández-Vara, and Mateas (2008, pp. 187-192) call this "challenge segmentation."

The opposite of a goal vector would be free roaming. NO MAN'S SKY (Hello Games 2016), for instance, has a goal vector that starts at the point where the player spawns for the first time. But this game possesses what is probably the vastest open world so far. According to one of its trailers (No Man's Sky 2016), NO MAN'S SKY offers the player an entire galaxy with 18 quintillion planets to explore. The other end of the goal vector is located at the center of this galaxy. Players, however, can take as much time as they please to reach the center or ignore this instruction entirely and focus solely on free roaming and exploration.

Nevertheless, not all unidirectional games that lack free roaming have goal vectors. A twist on the platformer genre is the *endless runner*, a genre popularized by games like CANABALT (Saltsman 2009), in which the character runs automatically and the player needs to press the jump button (or tap the screen when playing on a touchscreen device) in order to avoid hitting obstacles or falling through gaps. These games feature a camera that moves only in one direction, with the difference that both the frame and the avatar are in constant motion. Space in these games is even more analogous to a timeline than games like SUPER MARIO BROS. The objective in CANABALT and other endless runners is to get as far as possible, not to a particular point in space—just like with SPACE INVADERS, the CANABALT player is doomed to failure at some point. CANABALT prescribes a direction, but not an end goal, thus lacking a goal vector.

While SPACE INVADERS and CANABALT are centered on one objective type, it is common for games to combine more than one. The latest iteration of DOOM (id software 2016), for example, features a combination of goal vectors and areas where the objective is to defeat waves of enemies. Some sections have the player going from A to B, sorting platforming obstacles and killing enemies that stand in the way, but leaving enemies alive in these areas is a possibility. In this case, enemies are placed to impose a restriction (namely, avoid being killed by them). Other sections in DOOM function as small arenas that go into lockdown while demon hordes spawn and attack the player. Once demons stop appearing, the lockdown ends, and the player can proceed to the next section of the stage. In these sealed rooms, enemies are present both to place a restriction (do not get killed) and to set a goal (defeat them all).

Sometimes games also grant freedom to players to set their own goals in order to achieve an objective. DISHONORED, for example, allows players to approach their objectives peacefully while sneaking past guards. The central restriction while choosing this strategy is that the player character should not be detected by enemies patrolling the area. Alternatively, players can completely ignore the restriction of being seen and run into each area guns blazing, unleashing chaos and killing every opponent. The main restriction while playing this

way is losing hit points, which leads to the death of the player character. Both strategies can also be mixed to different degrees, allowing for creative playstyles.

One could object that there is no need for the differentiation between goals and restrictions. After all, the restriction of not losing all hit points, for example, could be phrased as the goal of keeping at least one hit point. While this is true, I chose to distinguish these two aspects of objectives because goals are events that happen because of player input, and restrictions can—and often will—occur even if the player remains inactive. Therefore, restrictions relate to the objectives of opposing players or computer-controlled entities.

GOING FORWARD

The typology introduced in this section describes different technical aspects and design features of video games that organize events in sequences. It is aimed at facilitating the analysis of video games in the present study, but also at benefitting the work of other scholars and game designers. In the following pages, the elements of this typology will be the standard vocabulary used to describe the formal aspects of the temporality of video games.

The medium of the video game evolves at an intense speed, and both the hardware and software used to develop and play video games are continually improving. Gaming consoles, for instance, are typically replaced by a newer model twice a decade, allowing games to grow in scale and complexity with each passing year. Independently of technological developments, game developers can also introduce revolutionary design features that change the landscape of gaming. Therefore, as stated at the beginning of this section, a typology such as this one should remain open for future amendments.