

Low G Man: Tomorrow

[Tentative Title]

Game Design Document

LGM: Tomorrow

Game Design Document

1. Introduction

Low G Man: Tomorrow (LGM:T) is a multi-level side-scrolling platformer in which the player, in the role of a robot designed to perform gravity-defying leaps, must battle a large army of space invaders. Armed with simple weapons and programmed to be the last bastion of defense, the protagonist must face dangerous enemies and complex environments, avoiding succumbing and destroying those behind the mysterious space assault.

The game will progress around a central narrative thread divided into a series of levels, presented to the player with a nearly linear structure. In each level, the player will have to use all the weapons in their possession to defeat the local "boss" and the hordes of henchmen he sends. The limited power of the protagonist's weapons will make life difficult for the player and his important mission.

LGM:T is designed exclusively for single player, and represents a continuation (actual or spiritual) of the Nintendo Entertainment System game of the same name.



1.1. Keyword

Platform, indie, old school, action, sci-fi.

1.2. Project philosophy

The game is the sequel to a NES title of the same name released in 1990. The hope of this project is to achieve an "HD remake" of the game that, while maintaining the core gameplay, offers new content and gives the overall experience a fresh feel. While retaining some common elements (such as the core gameplay, some enemies, and other references to the original game), the goal is to produce new content regarding the story, enemies, and a graphical style that, while incorporating appropriate references, appears to be 2.5D.

The decision to continue with the project is justified by the current historical moment, in which major indie productions remind us of a fairly large market for nostalgic games, with high skill requirements (especially reflexes and hand-eye coordination), and clear references to past successes. In this case, precisely because it isn't a success (certainly far from other contemporary titles), it has the opportunity to realize its potential, knowing that its simple rules (despite its highly interesting features) don't significantly differentiate it from other titles in the same genre.

License

The original game was developed in 1990 by Japanese software house KID and published by Taxan, also Japanese (but with an American division). KID declared bankruptcy in 2006 and sold its IP to a Korean company called CyberFront, which also closed a few years later (2012). Understanding who currently owns the IP (if any) can be advantageous for two key reasons. First, since it's an IP of little interest, it's possible that the current licensee will license it for negligible costs or even for free. Second, having a publisher/rights holder to talk to can lead to a collaboration that allows the game to be published with greater resources.

In any case, examining the issues related to the license is important to understand whether, from an IP point of view, it will be necessary to modify the contents to avoid plagiarism or whether, on the contrary, it will be possible to exploit them and, therefore, create a true sequel to the original game.

If the licensing issue cannot be resolved, we request that any direct reference to the original game be removed, starting with the title or the protagonist's "name."

Cameo

Regardless of the outcome of the issues described in the previous paragraph, the game must contain clear references to the universe we're drawing from: we therefore plan to include, throughout the game's levels, a series of cameos as references to monsters from the original game, or other 8-bit idols. The content must be carefully chosen after evaluating the original game and its main competitors.

1.3. Target platform

LGM:T is suitable for a platform capable of supporting a controller with at least 4 buttons. It's important to decide which platform to choose:

- **XBLA-PSN** This way, they would benefit from the enormous advantages associated with the flagship consoles, but without competing with AAA titles (and, therefore, the associated costs). The complexity of accessing these platforms remains to be assessed, both in bureaucratic terms (submission approval) and technological terms (dev console).
- **OUYA** Intended as a "reduced version" of the previous point, it presents some of its potential and minimizes some of its problems. A further advantage is the presence of some expertise (albeit very limited) on its guidelines and usability issues. A very careful evaluation of the market segment being addressed is necessary: choosing it as the sole platform may not ensure adequate diffusion of the product.
- **STEAM** In this case, a controller connected to the device is required, or a more in-depth study is needed to determine whether a keyboard is a suitable interaction tool to support gameplay. We suggest using this approach as a "Plan B" (or in parallel with one of the other proposed options), since it uses the most flexible platform. It's also necessary to evaluate the bureaucratic mechanisms that allow for Steam landing (Greenlight?).

1.4. Target Audience

LGM:T is aimed at a somewhat nostalgic audience, characterized by a strong passion for games with retro mechanics and a fair but challenging difficulty level. The dominant profile according to Bartle's taxonomy is that of the Achiever, which presupposes a strong focus on mastery, understood as improving one's skills and performance (focusing on speed and precision).

1.5 References

Walkthrough: <http://www.youtube.com/watch?v=IZfVJe59PTo>

2. Description of the game

The game is structured around two main scenarios: level selection and the game level.

The player begins their game experience within the first scenario, but the path ahead is blocked, requiring them to complete each level to proceed to the next. They must then choose a level (presumably the last one available, although they can replay any they've already completed) and, by doing so, enter the second scenario.

The game level is essentially a corridor (which extends bidirectionally, with a side view) in which the player must lead their character towards the final objective, be it the exit/conclusion point, a particularly difficult enemy to defeat, or something else. Each level is different in terms of its composition and, where applicable, the enemies involved.

The level's composition will require the character to defeat several enemies during his journey, and to find a practical way to overcome the environmental obstacles placed along the way (classic platforming).

Obstacles (animate and inanimate) within each level can cause damage to the protagonist, who can only tolerate a minimal amount of them (see his characteristics). Once the limit is exceeded, the player loses the game and is taken back to the beginning of the current level, or to the level selection if necessary. he prefers.

2.1. Level Organization

It's been said that there's a structure beyond the level structure, allowing the user to choose which level to tackle in detail. However, it's possible that this scenario (known as "level selection") is in turn organized into sub-sections called "worlds." This subdivision is more significant from a visual perspective than from a gameplay perspective, although it's conceivable that certain enemies could be associated with certain worlds.

The level selection scenario design should take into account the different game worlds and any bonus/extra levels, unlocked through separate mechanisms (see, for example, the extra levels of Super Meat Boy) and characterized by a particular graphic style or objectives.

3. Mechanics and dynamics

This chapter contains everything you need to know about the procedures and rules of the game, that is, what the player can and cannot do during the game.

3.1. Low G Man

Low G Man is the protagonist of the game and, in this role, represents the character that the player will move.

It is characterized by a series of basic parameters, which can be summarized as follows:

- **Hit points:** The amount of damage it can absorb before being destroyed. This value is assumed to be 4 like the original game.
- **Number of stasis bullets:** This is the number of bullets it can fire at once. Default is 1.
- **Spear length:** determines the maximum extension the spear can reach before it begins to retract (and, consequently, the overall time of the action). To be defined.
- **Jump height:** is the value that allows us to understand the intensity of the vertical component when the protagonist is asked to jump.
- **Movement speed:** determines how quickly horizontal movement can occur.

The parameters described above can be modified, during the game, by collecting power-ups that will increase (or reduce) their value.

Movement and jump

These are the character's most basic abilities. Essentially, they allow the player to control the character horizontally (forward and backward) and vertically. Horizontal movement, or walking, can be controlled directly. For vertical movement, the character will be subject to a fictitious gravitational force that will keep them on the lowest available plane (falling); however, the player can cause a jump of varying heights (depending on the character's characteristics and the duration of interaction with the button associated with the action). Jumping, like walking, is subject to the falling phenomenon, which keeps the character tethered to the supporting surface (therefore, this does NOT refer to completely free movement in two-dimensional space).

Pistol and spear attack

The character has two basic attacks at his disposal: a stasis pistol and a spear. These shouldn't be considered separate equipment, but rather an extension of the character, a dramatic representation of the protagonist's intrinsic abilities.

The gun allows the character to launch a horizontal projectile that is created at the protagonist's height (at the moment of interaction) and continues in a uniform (and straight) motion in the direction the protagonist is facing. The projectile's movement ceases only when it comes into contact with an enemy, an obstacle of some kind, or the edge of the screen, at which point it is destroyed and causes a different effect depending on the element it hits. With some exceptions, the impact of a projectile with an enemy has the effect of paralyzing it for a certain period of

time, rendering it harmless. Dynamics between the protagonist and the enemy that normally cause damage (see the enemy's "Contact Damage" ability) are suspended for the duration of the effect. When the stasis period ends, the enemy resumes its behavior from where it left off, without suffering any permanent effects. The pistol's bullets are infinite, but a certain rate of fire will be determined to prevent their uncontrolled production. While it cannot permanently eliminate an enemy, the pistol provides an immediate and short-term advantage in emergency situations and makes it easier to use the second type of attack, described below.

The spear is the main tool that allows the permanent elimination of an enemy. It's not very futuristic, but it's definitely effective: the protagonist can use it to injure and destroy enemies he encounters along the way. The weapon's strength, superior to that of the pistol, is offset by its limited use: it can only be used upward or downward. When the player performs the interaction associated with this attack, the spear projects in the desired direction (up or down), vertically relative to the character, for a length that varies depending on the protagonist's characteristics, usually passing over any obstacles or enemies. Once fully extended, the spear retracts, following the same path backward.

Colliding with a portion of the spear will cause the enemy to lose health; if the enemy's health runs out before the spear leaves contact (for example, by being retracted), the enemy is destroyed.

Attack with special weapons

Throughout the adventure, the character can collect a series of power-ups, including bullets for special attacks. These bullets, when selected, modify the gun's behavior to make it more lethal. The gun loses its ability to paralyze enemies, but gains the ability to actively damage enemies it hits.

There are several types of special weapons, which will need to be defined. Each will have a specific behavior that can significantly alter the mechanics (for example, changing the direction of the shot or the number of shots). Visual effort is also required to adequately represent these effects/transformations (in the original game, special weapons included fireballs, energy waves, boomerangs, and cluster bombs).

Additional movement?

Conceptually, the game offers enough mechanics to make the gameplay adequately varied. However, if the prototype analysis identifies too many similarities with the original game, we suggest implementing an additional action/procedure. It might be interesting to consider a move involving gravity, to reinforce the concept underlying the game (and embodied in the title itself). Examples include: throwing a grenade, quickly dodging, teleporting behind the enemy, briefly levitating, halving height, and so on.

3.2. Enemies

Despite their diverse nature and the diversity of their abilities, all enemies share basic characteristics similar to those of the protagonist, starting with the number of "hit points" they possess. While the concept is the same (a finite amount, once exhausted, the entity is considered defeated), their functioning is slightly different from that of the protagonist. Since the dynamics that define their damage, as already mentioned, depend on the amount of time during which the spear and the enemy intersect, hit points must be calculated based on the "mathematical" damage inflicted by the spear itself. To make this dynamic easier to imagine, one could imagine creating a "naive" association between these data, such as: "an instant of interaction between the spear and the enemy is enough," or "a full extension of the spear, when its length is basic, is enough to eliminate the enemy." The different nature of enemy hit points is also justified by the presence of the protagonist's special weapons, which in turn inflict direct damage.

Other common characteristics shared by all enemy types include movement speed (possibly zero) and a list of abilities (depending on the enemy). Given their varied nature and their "belonging" to a game world, this section aims to provide a macro-view of their main abilities, defining how they function so that they can be referenced later when creating enemies.

Damage on contact

This is a passive ability that determines one of the enemy's basic characteristics: if the protagonist comes into contact with it while it is not in stasis, they will receive damage. This dynamic will also be accompanied by an appropriate animation that will make the character recoil and cause a brief period of invulnerability, to avoid receiving multiple damage and being killed in a chain of "bounces."

Projectile attack

Allows an enemy to fire projectiles in a predetermined pattern, such as shooting straight ahead or in a specific direction (such as the character's location when the shot is fired). This action generates a projectile that continues in its initial direction until it hits an obstacle or the character (dealing damage, unless otherwise specified). Unless otherwise specified, enemy projectiles do not damage other enemies, only the player: if an enemy projectile collides with another enemy, the projectile simply continues along its path.

The visual representation of projectiles may vary depending on the enemy type.

Movement

This ability allows the enemy not to remain immobile, and to move following different types of rules.

In its most general sense, enemies will move in the opposite direction to the one the player is following to reach the game's objective, until they are destroyed or move far enough away from the character.

It's possible, however, that enemies could also be subject to gravity, or have more elaborate movement patterns (such as always aiming at the player, or moving back and forth, or in circles). It's hypothesized that this ability could be customized differently to make each enemy more unique.

Teleport

This ability allows the enemy to disappear and reappear in a different location than where they were previously. The ability can be used sequentially and follow a pattern, acting as an "instant movement." Both its frequency and the duration of the appearance/disappearance times can be configured.

Partial protection

An enemy with this trait is physically protected on one side. Any weapons the player carries (of any kind) will be ineffective on that side. To defeat the enemy, the player will therefore be forced to attack one of its unprotected sides.

It is also possible to set this side to be able to damage the character even while the enemy is in stasis, violating the general rule that an enemy in stasis temporarily loses its Damage on Contact ability.

Partial Protection can also apply to multiple sides at the same time, as long as at least one side remains uncovered (see, for comparison, Invulnerability and Total Protection).

The character design will make sure to use an appropriate metaphor, such as a shield, a pointed edge, and so on.

Total protection

Full Protection is a variant of Partial Protection that applies to all sides of the enemy, but is heavily timed to provide the player with a "window of vulnerability" that they can exploit to defeat the opponent.

Normally active, Protection will decay temporarily, at a configurable rate, and will remain deactivated for a preset period of time (also configurable), during which the enemy will behave as if it did not have this ability.

Invulnerability

An invulnerable enemy is one that cannot be destroyed: spears, special weapons, and any effects that can damage its health will have no effect on it. It can, however, be hit with a pistol, suffering the normal stasis effects (which the player can use to temporarily immobilize them, with all the advantages that entails).

Charge

Enemies with this ability can "target" the main character and charge at them in a charge attack. When this ability is activated, the enemy calculates the direction the character is currently facing. It then quickly moves in that direction, aiming to approach the player and damage them with another ability (Contact Damage, Partial Protection).

The direction is not changed after it has been calculated, meaning the player can move their avatar to avoid taking the hit.

Jump

This ability grants the jump action to enemies that have it.

Jumps can be used mindlessly (to complicate their movement pattern) or more creatively (through more elaborate AI, which takes into account the morphology of the terrain and allows enemies to exploit platforms and raised floors around them).

Stasis Immunity

An enemy immune to stasis is not affected by the stasis pistol. If hit by a stasis projectile, it is destroyed and its effects are ignored.

Stasis immunity does not protect against the effects of direct-damage weapons (such as the spear), which can still be used to damage and destroy those wielding them.

Stasis Immunity is incompatible with its counterpart, Stasis Vulnerability.

Vulnerability to stasis

Those with Stasis Vulnerability aren't affected by the normal effects of Stasis, but are instead directly damaged by the main character's pistol. Typically, one Stasis bullet will be enough to destroy them. This is therefore a weakness used to make certain enemies more unpredictable. It is obviously incompatible with Stasis Immunity.

3.3. Interactive environment

Some elements of the environment will be interactive, meaning that the protagonist will be able to interact with them to obtain a different effect depending on their nature.

Below are some “basic” items that can be predicted, and which, similarly to what was said for enemy abilities, can be chosen during the level design phase.

Fragile objects

Objects identified with this trait can be subject to various types of interaction by the main character (or other characters). For each object, it will be defined whether it is possible to break them (and, if so, with what weapons) and whether it is possible to move them (and, if so, whether they are subject to gravity).

These types of objects can be used in a wide range of situations: to hinder the player by inserting distracting elements, to assist them by creating alternative routes or blocking enemies, or to introduce puzzle-like dynamics that require the player to think carefully about how to create a viable path through the level. Level design must take their potential applications into account, identifying whether additional parameters need to be formalized.

Levers

Levers are an interactive tool that allows a character to modify something through the use of a level. They have a variety of uses: opening or deactivating barriers, allowing portions of the level

or fragile objects to appear, enabling or disabling specific features on other game elements (such as the movement of suspended platforms). Levers are an extremely versatile tool that allows the player to activate or deactivate the presence of game elements, change their state, or modify their behavior. A lever can be activated through a deliberate interaction performed in front of (or near) the lever itself.

The use of levers, as explained for breakable objects, can add a puzzle-like dimension to the game mechanics, introducing variety into the level design and requiring the player to take a less mindless approach to exploration within the level.

A variable, customized for each lever, will determine whether it can be reused or not after its activation (in the case of barriers or moving platforms this may make sense, but in cases where it is implicitly useless to re-execute the interaction, it will be prevented).

Pressure traps

Pressure traps basically work the same way as levers. Unlike levers, however, they are hidden (almost invisible) from the player's eyes and tend to have negative effects (such as spawning enemies, causing projectiles to appear, opening holes in the ground, and so on). They are activated automatically when the character steps on them. It's reasonable to expect enemies not to trigger pressure traps (though this should be configured for convenience).

Once activated, a pressure trap deactivates completely, making it impossible to activate again even if the character passes through the affected area again.