

ANALYSIS OF SKILL SYSTEM TESTING IN GAMES AND ITS IMPACT ON BALANCE

Introduction. In recent decades, the video game industry has undergone significant technological and conceptual transformation. One of the most notable developments is the widespread adoption of skill systems, often represented as skill trees, which have become a core gameplay mechanic across numerous genres. Originally skill trees were characteristic of role-playing games (RPGs). Nowadays they are commonly found in action-adventure games, strategy titles, platformers, and hybrid genres. These systems are instrumental in shaping player progression, influencing strategic decision-making, and sustaining long-term player engagement.

A skill system typically allows players to unlock, upgrade, or specialize abilities over time, thereby personalizing their gameplay experience. However, the increasing complexity and scale of modern skill systems create substantial challenges for developers, particularly in terms of testing and quality assurance. Poorly tested skill trees can result in gameplay imbalances, broken mechanics, usability issues, or unintended interactions between abilities. All this taken together may negatively affect player experience and lead to post-release patches or community dissatisfaction. Given the growing importance of skill systems in contemporary game design, the need for structured and systematic testing approaches has become increasingly evident.

Main material. An essential prerequisite for effective skill system testing is an understanding of skill tree architecture. From a structural perspective, skill trees can be broadly classified into two main categories: linear and non-linear.

Linear skill trees are characterized by a strictly sequential progression model, where each skill must be unlocked in a predetermined order. Access to advanced abilities is conditional upon the acquisition of preceding skills. This structure is often employed in games where developers aim to guide players through a carefully designed difficulty curve or narrative progression. Linear skill trees offer several advantages. The limited number of possible skill combinations reduces the complexity of testing, making it easier to identify errors, inconsistencies, or balance issues. Additionally, the predictability of player progression allows developers to simulate gameplay scenarios more efficiently. However, this increased flexibility substantially complicates the testing process, as developers must account for a wide range of possible interactions, dependencies, and edge cases that may lead to imbalances or system conflicts.

Non-linear skill trees, on the other hand, introduce branching paths and multiple development routes. They allow players to choose how their character evolves. In such systems, a single initial skill may lead to several alternative branches, each emphasizing different playstyles or strategies. This architectural approach is commonly used in modern RPGs and action games that prioritize player freedom, experimentation, and replay value. However, this increased flexibility substantially complicates the testing process, as every possible skill combination must be taken into account to ensure balance and prevent system conflicts.

Based on the analysis of skill systems implemented in modern games such as *Divinity: Original Sin 2*, *World of Warcraft*, *Hollow Knight*, and *Diablo III*, skill system testing can be divided into several interrelated stages. Each stage addresses a specific aspect of system reliability, balance, and usability.

1. Functional testing focuses on verifying the correct operation of individual skills. Each ability must be tested to ensure that it performs as intended, activates under the correct conditions, applies the appropriate effects, and displays accurate visual and textual feedback. Manual testing is inefficient since it takes a lot of time. Automated testing using scripts is a more suitable approach, especially when there are hundreds of skills. Automation enables rapid verification of skill behaviour after updates or balance changes, as well as reduces the likelihood of regression errors.
2. Integration testing evaluates how individual skills function when combined within the broader system. This stage aims to identify conflicts between scripts, unintended interactions between abilities, and dependency-related issues. (For example, a passive skill may incorrectly modify the effects of an active ability). Integration testing is particularly critical for non-linear skill trees, where players can unlock skills in various orders. Ensuring compatibility between all possible combinations is essential for maintaining system stability and fairness.
3. Balance testing is one of the most complex and important stages of skill system testing. Its primary goal is to ensure that no skill or combination of skills becomes disproportionately powerful or makes others irrelevant. Imbalanced systems can lead to dominant strategies, reducing player choice and negatively impacting gameplay diversity.
4. Usability and visual testing focus on the clarity, accessibility, and aesthetic coherence of the skill tree interface. This stage often involves creating and comparing multiple interface prototypes, as well as benchmarking against established solutions used in other games.

Conclusion. Skill systems have become an integral component of modern video game design. Their reliable implementation and testing constitute a critical development priority. A structured testing framework is essential for ensuring system stability, fairness, and player satisfaction.

An analysis of modern skill systems in video games, particularly skill trees, demonstrates their evolution from niche RPG mechanics to universal progression tools across many genres. Linear skill trees simplify testing thanks to predictable combinations, whilst non-linear ones, despite their advantages for player's freedom, significantly complicate the process due to multiple interactions and edge cases. Examples from games such as *Divinity: Original Sin 2*, *World of Warcraft*, *Hollow Knight* and *Diablo III* show that effective testing requires a phased approach: functional (with an emphasis on automation), integration (to identify conflicts), balancing (to avoid dominant strategies) and usability testing of the interface.

Overall, the increasing complexity of skill systems highlights the need for hybrid methods – a combination of automated scripts, simulations and iterative playtesting. This not only minimises the risks of imbalance and bugs, but also enhances long-term player engagement.

REFERENCES

1. A Survey of Video Game Testing [Electronic resource]. Available at: <https://ar5iv.labs.arxiv.org/html/2103.06431>
2. Effects of Action Video Game Play on Cognitive Skills: A Meta-Analysis [Electronic resource]. Available at: <https://tmb.apaopen.org/pub/qj0c4ij2/release/3>
3. Integration Testing [Electronic resource]. Available at: https://en.wikipedia.org/wiki/Integration_testing