EU VIDEO GAMES

TECHNICAL GUIDE

Creating educational video games without coding

Co-funded by the European Union

Akademia Humanistyczno Ekonomiczna w Łodzi

EGLE GAMIFICATION IN LEARNING AND EDUCATION

euphoria Net

TAVO EUROPA

sempre a frente

Project title & code

EU VIDEO GAMES (2024-2-PL01-KA220-YOU-000286883)

Project result name

WP2A2 Development of the Video Game Creation Facilitators: Tech Guide

Date of delivery

December 2025

Participating partners

Akademia Humanistyczno-Ekonomiczna w Lodzi (Poland, lead coordinator)

EGLE - European Gamification in Learning and Education (Belgium, activity leader)

Euphoria Net Srl (Italy)

Asociacija Tavo Europa (Lithuania)

Fundacja Sempre a Frente (Poland)

Table of contents

Chapter 1: Introduction 4

1.1 The purpose of this guide 4

1.2 Who is this guide for 5

1.3 How to use the guide 6

1.4 Video games: Brief overview and history 7

1.5 Why no-code games are useful in education 9

Chapter 2: Understanding video game creation 11

2.1 What is a video game? Key concepts and terminology 11

2.2 Examples of feedback in civic and social learning 17

Chapter 3: Game development fundamentals 19

3.1 Defining clear learning objectives and analysing the target audience 19

3.2 Selecting the appropriate game type 21

3.3 Designing core game mechanics 22

3.4 Integrating educational content seamlessly 25

3.5 Visual and auditory elements in educational games 25

3.6 Implementing feedback and assessment systems 28

3.7 Balancing challenge and accessibility 28

3.8 Creating engaging reward systems 29

3.9 Prototyping, testing and iterating 29

Chapter 4: Introduction to no-code game engines 31

4.1 Definition and benefits of no-code tools 31

4.2 How to choose the right game engine 34

4.3 A selection of accessible no-code engines 36

Chapter 5: Developing multimedia elements and mechanics 43

5.1 Visual and audio elements: Bringing your game to life 43

5.2 Designing for inclusion and accessibility 45

5.3 Game mechanics and multimedia logic 47

5.4 Playtesting, feedback and learning integration 50

Bibliography & further readings 53

Annexes 57

Chapter 1: Introduction

1.1 The purpose of this guide

The purpose of this guide is to **empower youth workers, educators, and young people** to explore the creative and educational potential of video games, particularly through **no-coding tools**.

No-coding tools are visual design platforms that allow users to create games without traditional programming, using drag-and-drop elements, logic blocks, or simple interfaces instead of writing code.

In today's rapidly evolving digital society, games are no longer limited to entertainment – they are **powerful means of communication, cultural expression, collaborative learning, and problem-solving**.

By engaging in game-making, participants develop not only technical awareness but also a wide range of **21st-century competencies** such as creativity, digital literacy, teamwork, and adaptability.

This guide is designed with **inclusivity and accessibility** in mind. Many people are discouraged from exploring digital creation because they assume that advanced programming skills are required.

By focusing on **no-code and low-code tools**, we remove these barriers and ensure that everyone - regardless of background, age, or technical expertise - can meaningfully participate in game-based projects.

The guide also aligns with the **European Commission's Digital Competence Framework (DigComp 3.0)**, which defines essential digital skills for citizens. In particular, it contributes to the areas of:

- **Digital content creation:** Enabling users to design interactive media and stories.
- **Safety:** Encouraging critical reflection about online interactions, privacy, and well-being.
- **Problem-solving:** Fostering creativity and resilience in approaching challenges.

Key goals of this guide

- **Empowering youth workers** to integrate game-based activities into their projects.
- **Providing young people** with hands-on opportunities to explore creativity and problem-solving.
- **Supporting inclusion** through no-code tools that lower entry barriers.
- **Encouraging critical engagement** with media and storytelling.

Tips & tricks

- ✓ **Start small**: Experiment with simple no-code platforms before exploring more complex tools.
- ✓ **Connect to interests:** Design activities that relate to music, sports, pop culture, or local stories.
- ✓ **Add value:** Link projects to broader educational goals, such as European digital literacy objectives.
- ✓ **Celebrate progress:** Focus on learning through experimentation, not only on producing a polished final product.

1.2 Who is this guide for

This guide is designed for a **broad and diverse audience** engaged in **education, non-formal learning, and youth empowerment projects**. Its **modular structure** makes it adaptable to different contexts, allowing each group of readers to take what is most relevant for their needs.

- **Youth workers:** Youth workers often seek **innovative methods** to engage young people in **creative, meaningful, and collaborative activities**.
  This guide provides them with **practical tools** to design workshops, youth exchanges, and community projects that use video games as a medium for **storytelling, critical thinking, and social interaction**.
- **Educators:** Teachers, trainers, and facilitators working in **schools or vocational training** can use the guide to bring game-making into **formal and non-formal education contexts**. **No-code tools** allow educators to **enrich digital literacy**

courses, ICT classes, and even **non-technical subjects** such as history, literature, or citizenship education.

- **Young people and learners:** The guide can also be used directly by **youth participants** who wish to explore **game design** and **digital creativity** on their own. While some sections target youth workers and educators, the **practical exercises and tool descriptions** are written in accessible language, making them suitable for **self-directed learning.**

For example:

- A youth worker designs a weekend workshop where participants create simple games to explore environmental challenges in their community.
- A teacher integrates a game-making exercise into a digital literacy class, asking students to design a small interactive quiz in Genially.
- A youth club runs a weekly no-code game jam, where young people collaborate in small teams to design prototypes inspired by their favourite music, movies, or local culture.

Tips & tricks

- ✓ **Identify motivation:** Find out whether participants are driven by fun, socialisation, creative expression, or learning outcomes.
- ✓ **Encourage peer learning:** Pair experienced participants with beginners to create a balanced learning environment.
- ✓ **Value the process:** Highlight experimentation, collaboration, and creativity rather than only the final product.
- ✓ **Adapt activities:** Tailor projects to local culture, group dynamics, and available resources.

1.3 How to use the guide

This guide is meant to be **practical and flexible**. It can be used in its entirety as a **structured manual** or **consulted in parts** according to the reader's needs.

Users are encouraged to choose their path based on their **context, role, and objectives**. Here are some recommended reading paths:

- **Educators:** Follow a structured path: **theory → examples → classroom application**.
- **Youth workers:** Focus on **practical tips, workshop outlines, and tool recommendations**.
- **Young participants:** Explore **case studies, examples, and tools** directly for inspiration.

Each chapter includes **"Tips & tricks"**, **practical examples**, and **suggested tools**, making it easy to adapt to **different learning environments**. Readers can also find **brief recaps** at the end of sections to highlight **key takeaways.**

Tips & tricks

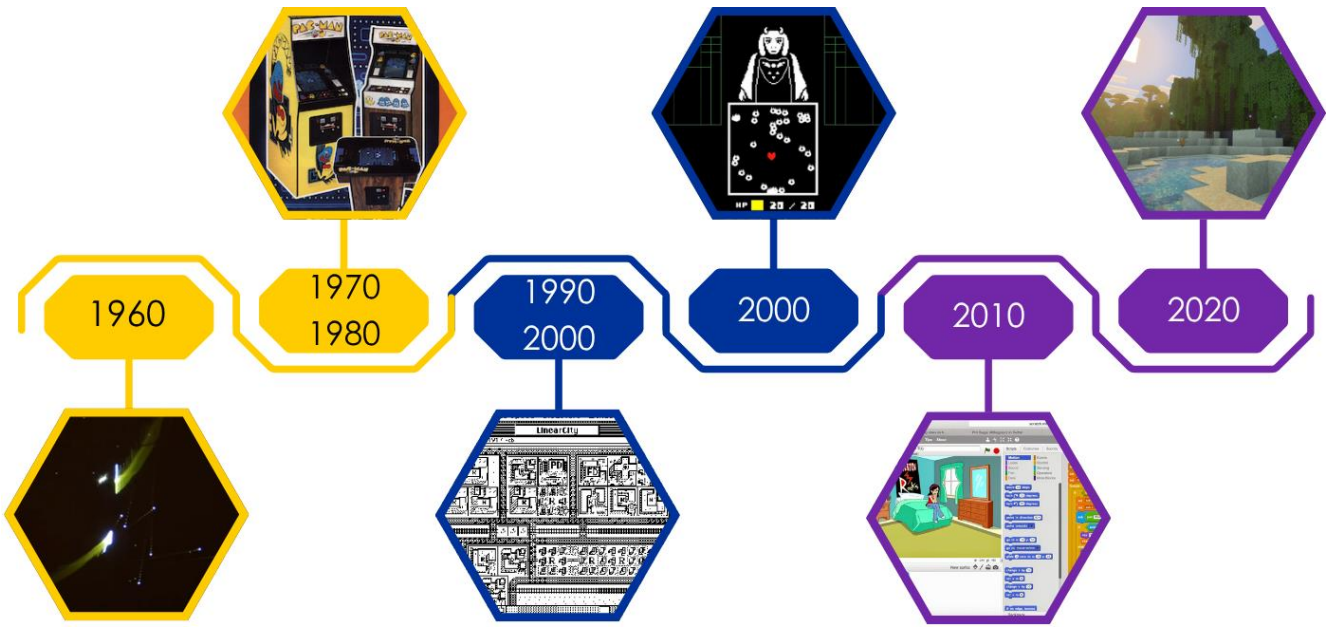
- ✓ **Start with your goal:** Identify whether you want to build technical skills, explore storytelling, or design a workshop.
- ✓ **Navigate selectively:** You don't have to read the guide linearly – jump to the sections most relevant to your project.
- ✓ **Document your process:** Take photos, screenshots, or short notes to reflect on progress.
- ✓ **Be flexible:** Each group and activity will be different – experiment, adapt, and iterate.

1.4 Video games: Brief overview and history

Understanding the **history of video games** helps explain their **educational potential**. Games have evolved from simple scientific experiments and arcade machines into **immersive platforms for creativity, collaboration, and learning**.
Over time, educators began experimenting with games in classrooms, leading to today's growing field of **game-based learning.**

Timeline of key developments

- **1960s – The birth of video games:**
  The first interactive electronic game, Spacewar! (1962), was created in a university lab at MIT for a science exhibition.
- **1970s–1980s – Emergence of arcade and home console games:**
  Classic titles such as Pong (1972) and Pac-Man (1980) became global phenomena. Early **educational games** like Number Munchers and The Oregon Trail introduced learning through play.
- **1990s–2000s – The rise of gamification in education and training:**
  Teachers and companies began recognising games as tools for **motivation, feedback, and engagement**. Popular classroom-friendly titles included SimCity (1989) and Carmen Sandiego (1985).
- **2000s onwards – The indie revolution:**
  Small teams began creating **creative, critical, and socially engaged games**, such as Braid (2008) and Undertale (2015).
- **2010s onwards – Democratisation of game creation:**
  No-coding platforms such as Scratch, Twine, and Construct made it possible for anyone to design interactive experiences.
- **2020s onwards – Expansion of immersive and accessible game-based learning:**
  Growth of **AR/VR and educational simulations**, with platforms like Roblox Education and Minecraft Education Edition. Increasing focus on **inclusivity and EU policy support** (e.g., Digital Education Action Plan 2021–2027).



Indie games have played a **key role** in showing how small teams – or even individuals – can create **meaningful and reflective works** that inspire discussion. The **no-coding movement** further extends this inclusivity, allowing **anyone to become a creator**, not just a consumer.

Tips & tricks

- ✓ **Use history as an icebreaker**: Invite participants to share their **first gaming experience**.
- ✓ **Show contrasts:** Display classics compared to modern examples, preferably of the same type or genre, to highlight evolution in design and storytelling. For example: Super Mario Bros (1985) vs. Celeste (2018), or Tetris (1984) vs. The Witness (2016).
- ✓ **Encourage discussion:** Ask participants what makes modern games more accessible or educational compared to older ones.

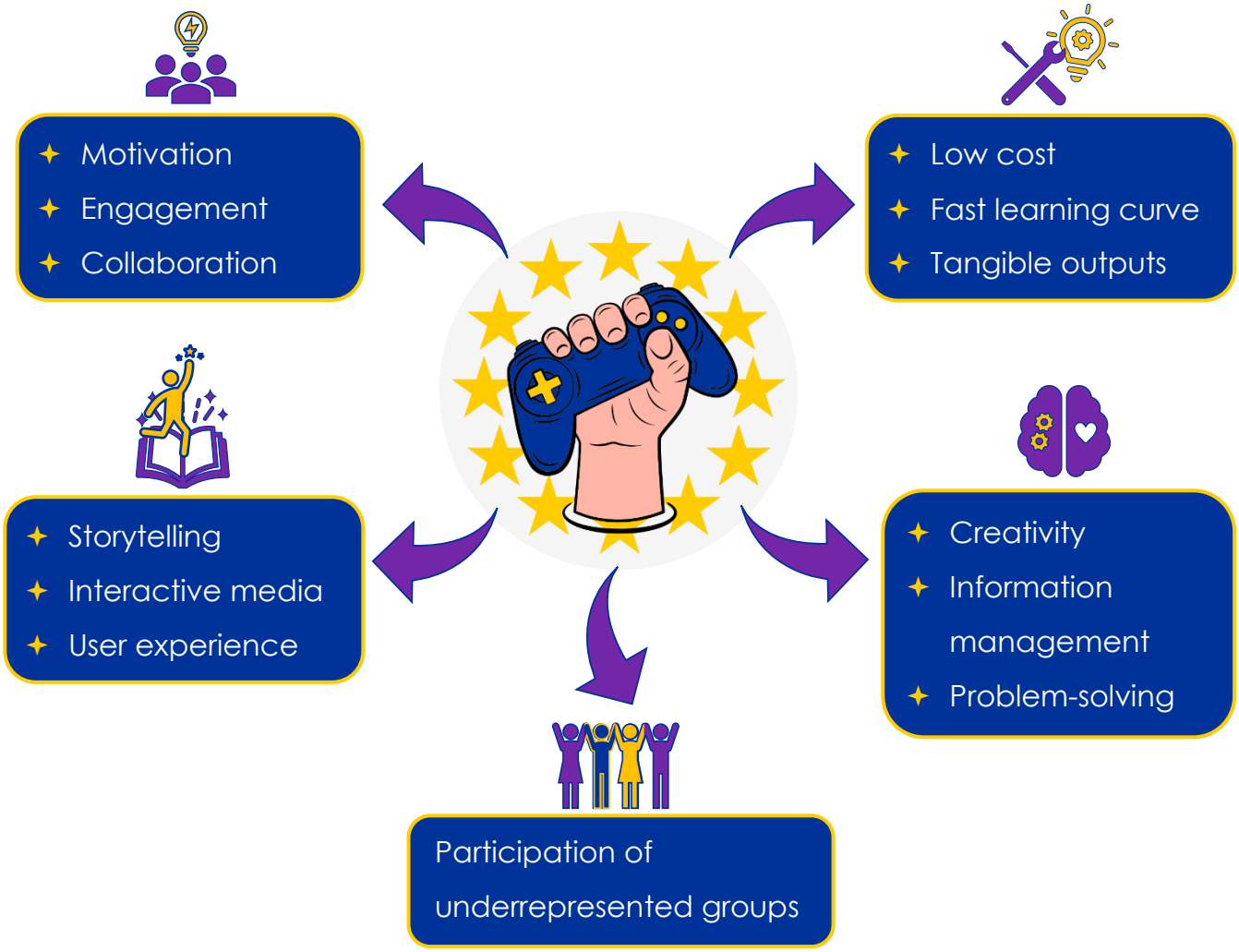
1.5 Why no-code games are useful in education

No-code games offer unique advantages in **education** and **youth work**. They **lower entry barriers**, making digital creation **accessible to everyone**, regardless of technical background or prior experience. These tools also support the development of a **wide range of digital and transversal skills** that are essential in today's society.

Main benefits

- **Game-based learning:** Increases **motivation, engagement, and collaboration** among learners. When students can design or modify games, they become **active creators**, not passive users – turning learning into an **interactive and meaningful process**.
- **Digital literacy:** Helps learners develop **creativity, critical thinking, and problem-solving skills**. Through design, players learn how digital systems work and how to express ideas through interactive media.
- **Media literacy:** Encourages reflection on how games communicate messages, shape narratives, and represent diversity. Learners gain insight into **storytelling, user experience (UX)**, and the **ethics of digital design**.

- **Inclusivity:** No-code tools make participation possible for **underrepresented groups** in tech, such as girls, young migrants, or learners with limited access to advanced technology. They promote **equal opportunities** for creativity and self-expression.
- **Practicality:** These tools are **low-cost**, have a **gentle learning curve**, and provide **tangible results** that can be showcased in classrooms, youth centres, or community events.



Tips & tricks

- ✓ **Start simple:** Use accessible tools such as Scratch, Twine, or Construct before moving to advanced editors.
- ✓ **Remix to learn:** Encourage participants to modify existing games to understand how mechanics work before designing their own.
- ✓ **Celebrate results:** Organise a small **game showcase** or demo day where learners can share and reflect on their creations.

Chapter 2: Understanding video game creation

2.1 What is a video game? Key concepts and terminology

Video games are one of the most influential cultural and educational mediums of our time. For many young people, they are not only a form of entertainment, but **a way to connect with others, explore ideas, and develop problem-solving skills**.

In the context of education and youth work, video games can become powerful tools to **stimulate creativity, encourage collaboration**, and introduce abstract concepts such as **democracy**, **citizenship**, or **sustainability** in an interactive way.

Understanding what a video game is and the key elements that make it work is essential for anyone who wishes to integrate games into non-formal education.

This section introduces the fundamental concepts and terminology that will help educators and youth workers recognise the structure behind every game and use it to create meaningful learning experiences.

Basic definitions

Video game	A digital interactive medium in which the player takes actions within a defined system of rules to achieve goals or experience a narrative. Games combine visual, auditory, and interactive elements to create meaningful engagement.
Player	The individual or group interacting with the game. The player makes decisions, reacts to challenges, and shapes the outcome through their actions.
Rules	The structure that defines what is and isn't possible within the game. Rules establish boundaries, set conditions for success or

	failure, and maintain balance between challenge and fairness.
Goals	The targets or objectives set for players, such as completing levels, solving puzzles, or achieving specific learning outcomes. Goals give purpose to the gameplay and motivate progression.
Progression system	The mechanism through which players advance in a game, by earning points, unlocking content, gaining experience, or improving skills. A well-designed progression system sustains engagement.
Prototype	An early, simplified version of a game used to test basic mechanics and design ideas before full development.
Playtesting	The process of having users (players) test the game to identify usability issues, technical bugs, or areas for improvement.
Beta version	A nearly complete game released for testing and feedback before its official launch.
Iteration	The practice of repeatedly testing, refining, and improving a game based on feedback and observations. Iteration is key to quality design.
Debugging	Identifying and fixing errors or issues (bugs) that affect how the game runs or behaves.
Storyboarding	A visual plan of the game's narrative or flow, showing scenes, levels, and player actions in sequence. Storyboarding helps clarify structure and pacing.
Sprites	2D visual elements or characters used in games, such as objects, players, or icons, often animated to create motion.
Script / scripting	The written code or dialogue that controls how a game behaves, including story events, character interactions, and gameplay logic. In non-coding engines, scripting often uses visual commands or drag-and-drop systems.

Trigger / event system	A setup that activates certain actions or changes in response to player input, for example, opening a door when a button is pressed or displaying a message when a level is completed.
Haptic feedback	The use of vibration or physical response to reinforce the player's interaction with the game, enhancing immersion and realism.
Accessibility	The practice of designing games so that people with different abilities can play and enjoy them. This includes visual contrast, subtitles, alternative controls, and simplified navigation.
Multimedia	The integration of text, images, sound, and animation to create an engaging interactive experience.

Common game genres

Understanding the main genres of games helps youth workers and educators select the most suitable type for their educational purpose.

RPG (Role-Playing Game)	Players assume the roles of characters in a fictional world, making choices that influence the story. Often includes dialogue, quests, and character progression.
FPS (First-Person Shooter)	Games that simulate action through the eyes of the player's character, emphasising reflexes and precision.
Platformer	Games where players jump and navigate between platforms or obstacles (e.g., Super Mario Bros).
Puzzle / logic game	Focused on problem-solving, critical thinking, and pattern recognition. Ideal for educational content.
Simulation	Games that mimic real-world activities (city management, farming, environmental systems) to teach systems thinking.
Adventure	Story-driven games where exploration and decision-making are central.
Strategy	Games that require planning and resource management, often teaching foresight and collaboration.

Serious game	Designed primarily for educational, social, or civic purposes rather than entertainment, often used in non-formal learning.

Core components

Game mechanics	The actions, behaviours, and control mechanisms available to the player (e.g., jumping, collecting, solving, building). Mechanics are the "verbs" of a game.
Gameplay	The actual experience of playing the game, resulting from the interaction between mechanics, rules, and player choices.
Narrative	The story or thematic context that gives meaning to the player's actions – such as exploring an ancient city, defending democratic values, or solving an environmental challenge. Narratives can be structured in two main ways: ◆ **Linear narrative:** The story follows a fixed sequence of events, where each player experiences the same storyline from beginning to end. This approach ensures clarity and is useful for games with a strong educational or moral message. ◆ **Branching narrative:** The story changes based on the player's decisions, offering multiple paths or endings. This structure encourages reflection and responsibility, as players experience the consequences of their choices – mirroring democratic participation and decision-making in real life.
Assets	The visual, audio, and multimedia elements that form the game's aesthetic (graphics, sound effects, music, animations).
User interface (UI) / User experience (UX)	The way players interact with the game and how information is presented to them (menus, scores, health bars, instructions).

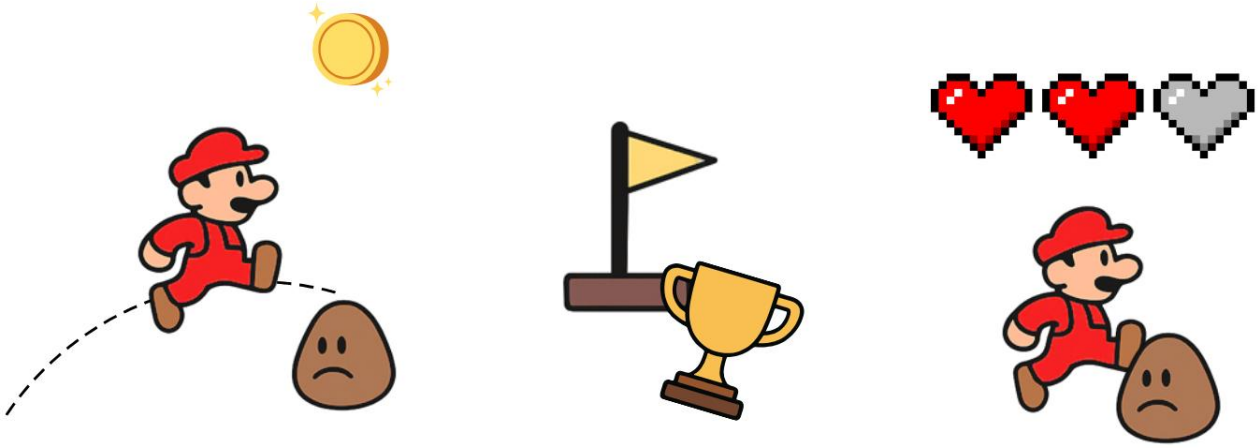
Tips & tricks

- ✓ Mechanics = what the player does
- ✓ Narrative = the story or context
- ✓ Assets = images, sounds, visuals
- ✓ UI/UX = how players see and use the game

Keep this glossary at hand when introducing game-based activities.

EXAMPLE OF GAME MECHANICS

In a platform game like Super Mario Bros



MECHANIC

Jump on enemies

Collect coins

GOAL

Reach the flag at the end of the level

RULE

If the player touches an enemy without jumping on it, they lose a life

This simple structure demonstrates how **mechanics**, **goals**, and **rules** interact to create gameplay. Each of these elements can also be linked to **educational and democratic learning goals**.

For example: If this mechanic were adapted to a civic education context:

- **Mechanic:** Players overcome obstacles by collaborating or sharing resources (symbolising dialogue and solidarity).
- **Goal:** Work together to reach a common destination (representing collective decision-making).
- **Rule:** If a player ignores cooperation, progress slows down or becomes impossible (illustrating the consequences of exclusion or lack of participation).

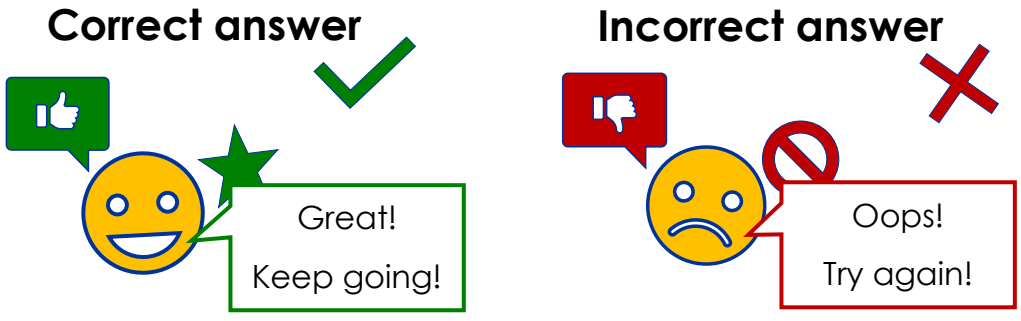
In this way, gameplay becomes an **analogy for democratic processes,** players learn through experience that participation, empathy, and collaboration lead to success, while isolation or lack of engagement leads to failure or conflict.

Supporting concepts

- **Levels / Stages**: Segments of the game with their own goals and challenges, often increasing in complexity.
- **Multiplayer / Single player**: Games designed for one player versus those allowing collaboration or competition among several.
- **Feedback systems or "Feedback loops"**: How the game responds to player actions (e.g., sound when an item is collected, message when a task is completed).

WHY FEEDBACK MATTERS

Imagine a quiz game for civic education



Feedback helps learners understand their progress and stay motivated.

2.2 Examples of feedback in civic and social learning

1. Collaborative decision-making

In a game where players act as members of a city council or youth parliament:

- When players **reach consensus**, a visual indicator (like a city growing or people celebrating) shows collective success.
- If they **fail to cooperate**, the environment stagnates or public satisfaction decreases.

This feedback mirrors real democratic processes – showing that collaboration and dialogue lead to better outcomes for the community.

2. Community problem-solving

In a sustainability or urban-planning game:

- When players invest in eco-friendly policies, the environment becomes greener or pollution levels drop.
- Poor decisions result in visible consequences, such as waste buildup or unhappy citizens.

Feedback here teaches young people **the impact of civic choices** and the importance of accountability and long-term thinking.

3. Digital Citizenship and Media Literacy

In a media simulation game:

- Players who correctly identify misinformation receive informative feedback explaining why a source was unreliable.
- Those who spread false information see negative effects, like decreased public trust or chaos in the virtual society.

This reinforces **critical thinking, responsibility, and ethical communication**, all key aspects of democratic participation online.

Why this matters in education

Understanding the basic components of a video game is more than just learning technical language: it provides educators and youth workers with **the foundation they need to design meaningful activities**. When a youth worker recognises how mechanics shape player behaviour, or how feedback keeps motivation alive, they can intentionally adapt these elements to promote skills such as **collaboration**, **decision-making**, or **critical thinking**.

For example, a simple rule in a game can be used to simulate democratic processes, showing young people the consequences of fair play, participation, or collective choices. Similarly, narratives can be designed to reflect social issues, allowing players to step into roles that make them experience perspectives different from their own. Even the visual and sound assets are not only aesthetic features but can be carefully chosen to make the game inclusive and accessible to all learners.

By mastering these key concepts, **educators move beyond seeing video games as "just play"** and start to recognise them as structured environments where learning can happen. In this sense, every mechanic, every piece of feedback, and every story element becomes an opportunity to foster civic awareness, empathy, and digital literacy, skills that are central to active citizenship in the 21st century.

Tips & tricks

When designing a game activity with young people:

- ✓ **Start simple** (one mechanic, one rule, one goal).
- ✓ **Use familiar stories or contexts** (school, local community, daily life)
- ✓ **Focus on clarity** of instructions rather than complex graphics.

Chapter 3: Game development fundamentals

3.1 Defining clear learning objectives and analysing the target audience

Begin by identifying exactly what learners should know or be able to do after playing your game. Learning objectives should be **specific, measurable, and aligned with educational standards** when applicable.

Tips & tricks

At first, ask yourself these key questions:

- ✓ What specific **knowledge or skills** should players acquire?
- ✓ How will you **assess** whether learning has occurred?
- ✓ What depth of understanding is required (**memorisation, application, analysis**)?

Identifying learning objectives before designing game mechanics is crucial for creating effective educational games.

When you start with clear objectives, you're defining exactly what understanding or behaviour change you want players to achieve. For democratic values specifically, you need to know whether you're teaching concepts like **majority rule, minority rights, deliberation, compromise, or civic participation**.

Without this clarity, you risk creating a fun game that doesn't actually teach anything meaningful, or worse, accidentally reinforces misconceptions about how democracy works. Starting with objectives also helps you measure success.

If your goal is for players to understand that democratic decisions require balancing competing interests, you can later assess whether your game actually achieves this.

How to match objectives to core mechanics

The key is making the mechanic embody the concept, not just illustrate it. The gameplay itself should require players to practice the democratic value you're teaching.

For example: If your objective is teaching negotiation and compromise, your core mechanic might require players to trade resources or votes to build coalitions - they experience compromise through necessity, not through a lecture. If you're teaching about information literacy in democracy, the mechanic could involve players weighing conflicting sources before making decisions, where trusting unreliable information has consequences.

Tips & tricks

Match the mechanic to the cognitive or social skill involved.

- ✓ Teaching about representation? Have players advocate for different group interests.
- ✓ Teaching about checks and balances? Create interdependent player roles where no one can act unilaterally.

The tighter the connection between what players do mechanically and what you want them to learn conceptually, the more powerful the educational experience becomes.

Thus, understanding your players is crucial for **creating appropriate challenges** and engagement strategies. Different age groups and profiles have varying cognitive abilities, attention spans, learning styles and motivations.

Tips & tricks

At the beginning, you should consider:

- ✓ **Prior knowledge:** What do players already know about the topic of democracy in general?
- ✓ **Gaming experience:** Are they casual players or experienced gamers?
- ✓ **Context of play:** Will they play in classrooms, at home, or on mobile devices?
- ✓ **Accessibility needs:** Consider diverse learning styles and potential disabilities. Social and cultural sensitivity, focused on the needs of different audience groups, is essential to ensure all players can fully participate.

To fulfil this step in the best way, try to build **learner personas** to keep your audience in mind throughout the whole process of development, which will also be useful for the next steps regarding the game type to focus on.

For example: Maria, age 15, loves puzzles, struggles with mathematics and often plays games on her phone. The type of game and challenges she could efficiently and positively respond to would need to take those elements of her profile into account: investigation and logic games, limited use of calculations, playable on a mobile device.

Following this method, you can prepare several fairly detailed descriptions of potential users of your game in the format you prefer. You can quickly find simple templates for creating personas, such as on **Canva**, and present them as cards that you can refer to throughout the design process to ensure that your game fits the needs, styles, and profiles of the targeted players.

3.2 Selecting the appropriate game type

The game genre should naturally support your learning objectives rather than work against them. Different game types excel at teaching different things.

Most common genres and learning matches

Simulation games	Systems thinking, cause-and-effect relationships, strategic planning
Puzzle games	Logical reasoning, pattern recognition, data analysis
Adventure and role-playing games	Narrative comprehension, decision-making, exploration of complex topics
Quiz games	Fact recall, speed of recognition, competitive knowledge testing
Strategy games	Resource management, long-term planning, analytical thinking

For example: For teaching democracy and identifying civic values, a simulation where students vote on real classroom choices (field trip locations, project topics) to practice direct democracy would be more engaging and efficient than a simple quiz or word puzzles.

3.3 Designing core game mechanics

Game mechanics are the rules and systems that define player interactions. In educational games, mechanics should create meaningful practice opportunities for the target skills. In the following table are some examples.

Game type	Learning outcomes	Specific mechanic(s)	Description and guidance
Simulation games	▪ Systems thinking ▪ Cause-and-effect relationships ▪ Strategic planning	Resource management Policy/law creation	Players manage civic resources (public trust, tax revenue, popular support). Cause-and-effect is reinforced by implementing policy/law creation processes. **For example:** A "Free Press Act" policy increases public trust but decreases immediate revenue, requiring strategic planning to balance the system.
Puzzle games	• Logical reasoning • Pattern recognition • Data analysis	Rule-set puzzles Drag-and-drop Association Dialogue enigmas	Rule-set puzzles model democratic processes and direct players through civic concepts. **For example:** The player must place "citizen interest blocks" into a "legislative grid" following rules (the pattern) that satisfy multiple, conflicting constituencies (the logical reasoning challenge of compromise).
Adventure games	▪ Narrative comprehension ▪ Decision-making ▪ Exploration of complex topics	Dialogues Quests and tasks Exploration	Exploration involves traversing different areas to understand diverse social issues (complex topics). Dialogues with NPC (Non-Player Character) use a persuasion or rhetoric system where player choices (decision-making) are judged by logic,

			emotion or ethical appeal. Quests are structured as civic actions. **For example:** Players must explore a neighbourhood and mediate a dispute that involves complex dynamics between citizens.
Quiz games	▪ Fact recall ▪ Speed of recognition ▪ Competitive knowledge testing	Timed fact recall Debate battle	Timed quizzes test fact recall of constitutional amendments, historical democratic milestones or voting procedures. A unique mechanic, debate battle, uses speed and accuracy in recognising evidence to win arguments against opposing viewpoints, framed as a competitive knowledge test.
Strategy games	▪ Resource management ▪ Long-term planning ▪ Analytical thinking	Item collection Combat	Resource management tracks budgets, items, political capital or public opinion. Item collection involves gathering essential materials, evidence (data) or support (votes) over multiple turns (long-term planning). "Combat" is reframed as debates, elections or influence campaigns, where success is determined by the analytical deployment of collected resources to sway key demographics.

3.4 Integrating educational content seamlessly

The most effective educational games make learning feel like a natural part of the experience rather than an interruption.

Tips & tricks

Here are the most useful integration strategies:

- ✓ **Embedded learning**: Weave educational content into the game world and storyline. In a game about the history of democracy, players may need to understand the reasons behind the emergence of democracy in order to effectively develop the systems necessary for it to function (e.g., the agora), making historical knowledge directly useful.
- ✓ **Contextualised challenges**: Develop educational tasks within meaningful scenarios. Instead of "list social inequalities," propose the challenge of creating characters and situations in which these inequalities occur. This is a motivating starting point for finding solutions to remedy these inequalities.
- ✓ **Discovery-based learning**: Let players experiment and discover the rules through gameplay. Playing parliament can allow players to build political parties that defend the interests of particular groups. Players learn to observe what happens when coalitions are formed and what the role of the opposition is. They learn about the principles of democracy through trial and error.

3.5 Visual and auditory elements in educational games

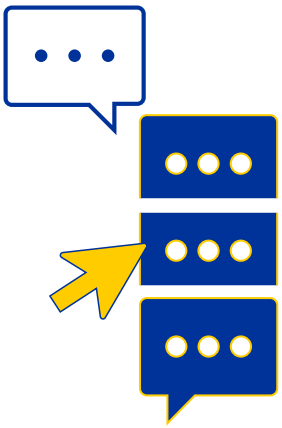
Achieving impact with an educational game, especially one focused on abstract concepts like democratic values, relies heavily on thoughtful design, not only in narrative and mechanics, but also in visual and auditory domains.

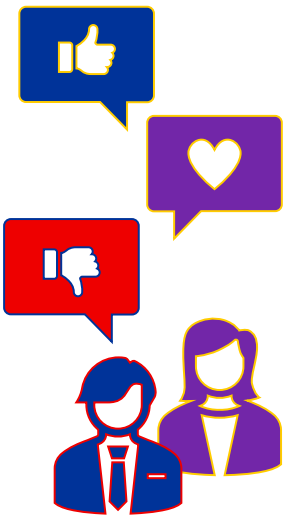
Visual clarity and consistency

Visual design must serve function, especially when teaching complex values.

Creating visual hierarchy (interactive vs. decorative)

Players must instantly understand what they can and should interact with. Interactive elements (like buttons to draft a bill or dialogue options) must be clearly distinguished from decorative background elements (like city scenery or static character portraits). This prevents cognitive overload and ensures the player focuses their limited attention on the core mechanisms of democratic engagement. Interactive components should be highlighted by colour, size, or animation.



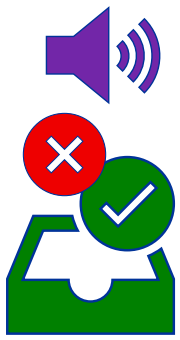


Maintaining consistent visual language

A consistent style, from iconography for resources (e.g., all "public trust" icons look similar) to the colour palette used for different political factions, reduces confusion and makes the game world legible. Consistent design builds trust and professionalism. If one faction is blue throughout the game, and all negative policies are red, this visual language helps players quickly process information and make informed decisions, mirroring the need to consistently interpret information in real-world civic life.

The power of sound

Sound is a crucial, yet often underestimated tool in the educational game developer's arsenal.



Providing feedback

Auditory cues immediately confirm player actions, which is vital for learning. In a game about voting, the sound of a successful ballot submission ("ding!") provides positive reinforcement, while an error sound guides the player to correct a mistake, ensuring they internalise the correct procedure.

Creating an atmosphere

Sound design can immediately set the tone. For a serious political simulation, a low, ambient hum and dramatic, non-intrusive music can convey the gravity of decision-making. For a more light-hearted civic engagement game, upbeat, orchestral music can encourage participation and approachability. This atmosphere keeps the player immersed in the topic.

Guiding the player's attention

Sound directs the player's focus without cluttering the screen. A sudden audio spike can signal a critical event, such as a news alert about a change in public opinion, drawing the player's eye to the relevant UI element and reinforcing the importance of being reactive in a democratic system.

3.6 Implementing feedback and assessment systems

Effective feedback is what transforms play into learning. Your game should provide multiple forms of feedback that guide players toward mastery.

Here are some feedback types:

- **Immediate feedback**: Players should instantly understand the consequences of their actions. Visual and audio cues confirm correct choices and gently redirect incorrect ones.
- **Explanatory feedback**: When players make mistakes, explain the underlying concept rather than just marking answers wrong. Adding "Remember, the separation of powers is one of the key principles of democracy" teaches more than simply stating "Incorrect."
- **Progress feedback**: Show players how far they've come and what they've mastered. Progress bars, skill trees, and achievement systems provide motivation and a sense of advancement.
- **Adaptive systems**: Track player performance to identify struggling areas and adjust difficulty accordingly. If a player consistently fails at a particular concept, provide additional practice or simplified versions before moving forward.

3.7 Balancing challenge and accessibility

Finding the right difficulty balance keeps players in the "flow state" – engaged but not frustrated, challenged but not overwhelmed.

Some balancing techniques include:

- **Adjustable difficulty**: Offer multiple difficulty modes or let the game adapt automatically based on performance. Some players may need more support, while others seek greater challenges.
- **Scaffolding**: Provide hints, tutorials, and support systems that players can access when needed. Make help optional so it doesn't interrupt players who don't need it.
- **Multiple paths to success**: Allow different strategies and approaches to solve problems. This accommodates various learning styles and keeps gameplay from

feeling rigid. It is also a very good, inclusive and confidence-building strategy for players with different difficulties and different levels of prior knowledge.

3.8 Creating engaging reward systems

Rewards motivate continued play and provide psychological satisfaction for learning achievements.

Basic effective reward strategies include:

- **Intrinsic rewards**: The best rewards come from the gameplay itself - the satisfaction of solving a puzzle, unlocking new content, or mastering a skill. These create lasting motivation.
- **Extrinsic rewards**: Points, badges, and leader boards can boost engagement when used thoughtfully. Ensure these don't overshadow learning objectives or encourage gaming the system rather than genuine understanding.
- **Meaningful rewards**: Connect rewards to learning progress. Unlock new game features or content as players demonstrate mastery, reinforcing that learning leads to new opportunities.

3.9 Prototyping, testing and iterating

No educational game is perfect on the first attempt. Testing with real players reveals what works and what doesn't. Always test your games on the specific target group for which you intended them. Comments and suggestions from outside this group may be helpful, but they are not reliable feedback on the assumptions and ambitions of your project.

The following infographic proposes essential steps of the iterative process.

Basic prototyping

Before investing in development, test core mechanics with a basic frame of your game or digital mock-ups. This identifies fundamental design issues early.

Playtesting with the target audience

Observe young people within your target audience playing your game. Watch for confusion, frustration, or disengagement. Ask players to think aloud as they play.

Learning assessment

After playtesting, discuss with the players to assess whether they learned the expected content. This validates your educational effectiveness beyond engagement metrics.

Iterative refinement

Use testing feedback to refine mechanics, adjust difficulty, clarify instructions, and improve learning integration. Great games emerge through several rounds of iteration.

Tips & tricks: Common pitfalls to avoid:

- ✓ **The quiz trap:** Simply presenting educational questions between unrelated gameplay segments creates a disjointed experience. Aim for seamless integration.
- ✓ **Overcomplication:** Adding too many mechanics or features can overwhelm players and obscure learning objectives. Start simple and add complexity purposefully.
- ✓ **Ignoring fun:** An educational game that isn't enjoyable won't be played. Prioritise engagement alongside learning effectiveness.
- ✓ **One-size-fits-all:** Different learners have different needs. Build in flexibility and personalisation where possible.

Chapter 4: Introduction to no-code game engines

4.1 Definition and benefits of no-code tools

No-code game engines are software platforms that allow users to **create games and digital experiences without programming**. These tools replace complex coding languages with **intuitive interfaces and pre-built elements**. Instead of typing code, users work with **visual elements, flowcharts, drag-and-drop components, dropdown menus and conditions in plain language**.

The rise and importance of the no-code movement

The no-code movement emerged in the early 2000s to **bridge the digital divide between programmers and amateurs**. It gained significant momentum in the 2010s, driven by the recognition that **creativity shouldn't be limited by technical skills**. Indeed, many innovative ideas come from people who aren't experts in the matter and who may not know how to bring those ideas to life.

By removing these barriers, the no-code movement has **democratised digital creation**, enabling millions of people to **build websites, mobile apps, games, and digital experiences** that were previously only accessible to those with years of programming training. With no-code engines, youth workers can easily guide youth through game design and **keep the focus on educational objectives, creativity, practical skills and meaningful engagement**.

The benefits of no-code tools for youth involvement

◆ Empowering youth without programming barriers

Many young people, particularly those from underrepresented groups in technology, such as women and people of colour, may feel that game creation is "not for them", especially when faced with **complex and intimidating coding**.

Intuitive interfaces with immediate visual results enable young people from all backgrounds to **develop a sense of confidence and agency in digital spaces**, allowing them to see themselves **not just as consumers but as creators who can shape digital experiences** for others.

Tips & tricks

- ✓ **Start with simple projects to build confidence:** Focus on the ideas rather than completing a full result. Explore prototypes and emphasise creativity and perseverance rather than technical perfection.
- ✓ **Show examples of games made by peers:** Select games that have been created by young people with the same engine your audience will use, so they can clearly see the possibilities and potential of their ideas and goals.

◆ Fostering creativity and critical thinking

When designing games, young people must consider how to **break content down into smaller tasks** and make their ideas **clear and engaging**. This process develops their ability to **analyse complex topics and communicate effectively**. The iterative nature of game design – **testing, receiving feedback, and refining** – also mirrors the **critical thinking processes** essential for civic engagement and democratic participation.

In practice: When young people create video games, they learn how to:

- ➢ Make abstract and complex concepts accessible.
- ➢ Break down complex topics and analyse key elements to emphasise.
- ➢ Represent and navigate cause-and-effect relationships.
- ➢ Imagine a variety of potential outcomes and decision paths.
- ➢ Acknowledge criticism and feedback to update and improve their work.

◆ Promoting digital literacy and problem-solving

Working with no-code engines develops **essential digital literacy skills, such as logical thinking**, and helps young people understand how digital systems work without complex programming. These tools introduce concepts such as **variables or**

conditional logic in an accessible manner. The problem-solving skills developed by **debugging issues, optimising user experiences, and iterating based on feedback** are transferable to many other areas of life.

For example: Through game creation, a young person might learn about:

- **"If-then" logic** by creating a quiz game about European history.
- **Data management** by tracking progress through a democracy simulation.
- **Event-driven programming** by designing dialogue systems for debates.
- **Variables** by creating voter point systems or progress tracking.

Tips & tricks

- ✓ **Start with simple mechanics:** Explore basic functionalities such as "click to continue" or multiple-choice questions, and gradually introduce more complex logic, such as scoring systems or branching paths.
- ✓ **Use real-world analogies to explain digital concepts:** Represent specific mechanics and interactions as similar to existing and common realities, such as voting systems, rewarding efforts, and logical cause-and-effect.
- ✓ **Encourage experimentation and trial-and-error:** Allow designers to "break" their work to test their limits, identify flaws and progress beyond their ideas.

◆ Making learning interactive and engaging

Video games transform passive learning into **active, engaging experiences**, allowing youth to retain information more efficiently than with traditional methods. When young people create educational games, they must have a **deep understanding** of the subject matter, reinforcing their own learning while **developing empathy for different learning styles and perspectives**.

In practice: The benefits of game-based learning:

- Makes complex subjects approachable and memorable.
- Reinforces the creator's own understanding of the topic.
- Provides immediate feedback and potential pathways.
- Accommodates different learning preferences.
- Enables personalised educational experiences.

For example: Instead of reading about democracy, youth can create:

- **Voting simulation games** where players experience different electoral systems and the way they function in real life.
- **Policy-making games** that show the complexity of reaching consensus.
- **Historical scenario games** that explore key moments in European integration and the birth and evolution of the systems we know today.
- **Role-playing experiences** where players represent different constituencies.

Tips & tricks

- ✓ Keep initial learning goals **simple and clear**.
- ✓ Include **multiple ways to succeed** or explore content.
- ✓ Provide **meaningful feedback**, not just points or badges.
- ✓ **Test** games with the target audience and **iterate** based on feedback.
- ✓ Connect game experiences back to **real-world applications**.

4.2 How to choose the right game engine

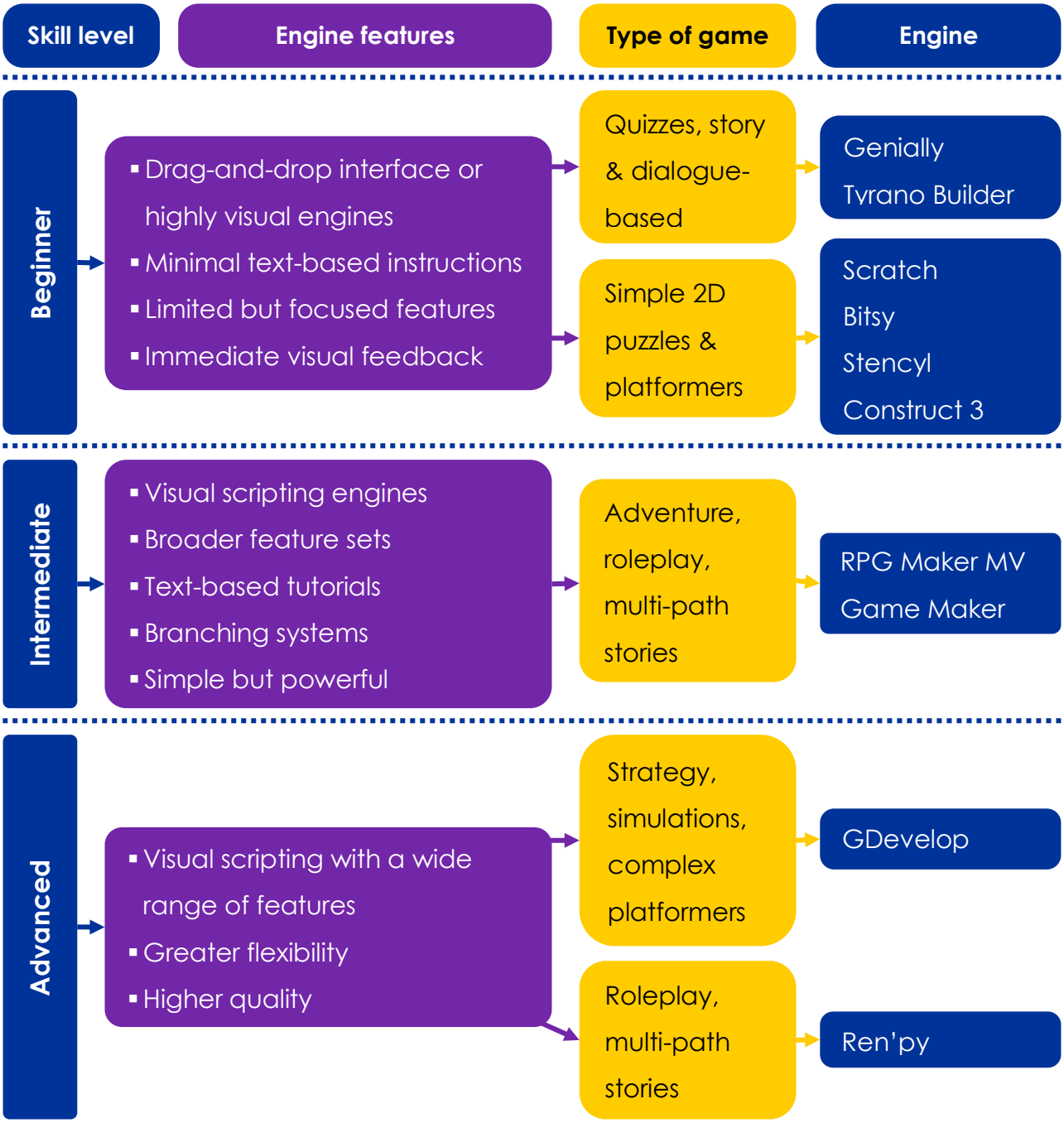
Selecting the appropriate no-code game engine is crucial for success, as the wrong choice can lead to **frustration, abandoned projects, or missed learning opportunities**. To identify which game engine is best suited for your needs and your audience, consider the following aspects:

Assessing difficulty level

- ☐ Can participants navigate basic computer interfaces confidently?
- ☐ Have they created digital content before (such as videos)?
- ☐ What's their attention span for learning new tools?
- ☐ How do they handle technical difficulties?

Recommendations for each technical level

Once the difficulty level is established, identify the type of game you wish to create in order to select the right engine. Each of the suggested engines will be described either in this chapter or in the annexes.



Tips & tricks

- ✓ **Plan for testing time** (25% of development time) and budget for the learning curve on the first projects.
- ✓ Adapt the length of the sessions and include **regular breaks or varied steps**, depending on the participants' age or potential learning disorders.
- ✓ Make sure that the computers being used to create video games have **adequate specifications** (for example, RAM, processor, storage, etc) so that the software can run smoothly. The engines we recommend don't have high requirements or particular needs, but certain devices may need adjustments to ensure an adequate experience and avoid technical issues.

4.3 A selection of accessible no-code engines

Based on research and prior experience in game-based projects, we have selected six game engines that best serve educational purposes whilst remaining accessible to non-programmers. Each engine was chosen for its strengths and ability to support different types of educational projects.

Tips & tricks

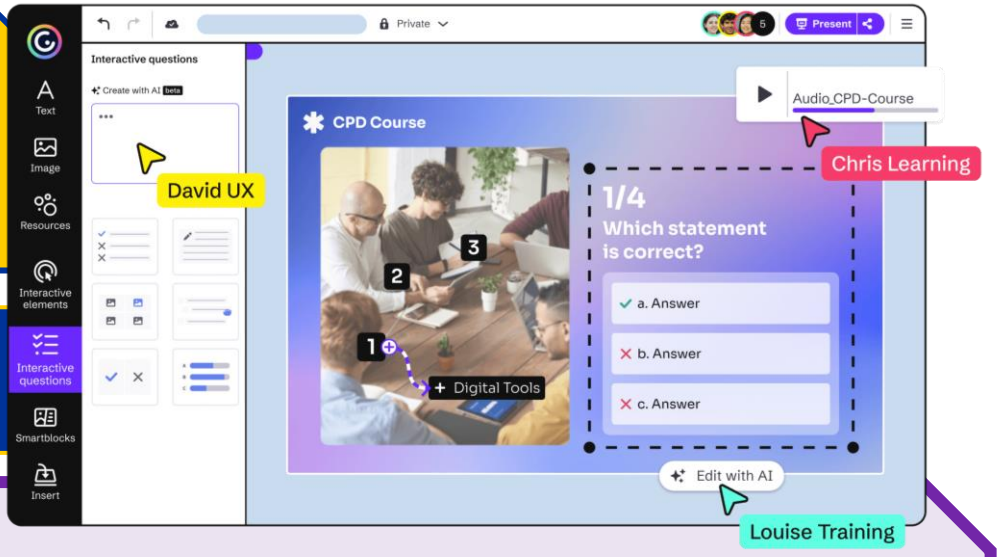
- ✓ Choose engines that **match your technical constraints** (hardware, time, user experience) rather than the most powerful available option. **Successful simple projects outperform abandoned complex ones**.
- ✓ Before selecting and proposing a game engine to young people, make sure to **explore its functionalities** properly and **discover a variety of games** created with that engine. Familiarise yourself with it properly and **identify the potential pitfalls and difficulties** your audience may encounter so you can adequately address and manage them.

Genially

Interactive visual presentations

Difficulty level:

Beginner ★☆☆☆☆



Technical features

- Browser-based platform (no installation)
- Interactive presentation creator with gaming elements (referred to as "PowerPoint on steroids")
- Template library for visuals and formatting
- Multimedia integration (videos, audio, images, animations)
- Branching pathways through clickable elements
- Real-time collaboration features
- Wide range of features through S'CAPE tools: drag-and-drop, enigmas, timers, notes, fill-in, etc.

Best for

- Interactive dialogue
- Interactive maps
- Decision-making
- Visual quizzes

Strengths

- Intuitive drag-and-drop interface
- Professional results quickly achievable
- Immediate visual feedback
- Easily shared

app.genially.com

Tyrano Builder

Visual novel with visual interface

Difficulty level:

Beginner ★☆☆☆☆

Technical features

- Complete visual interface (no scripting required)
- Component-based design using drag-and-drop
- Live preview system for immediate feedback
- Built-in character and background management
- Audio integration with music and sound effects
- Export to multiple platforms, including web and mobile

Best for

- Beginner-friendly interactive stories
- Decision-making scenarios
- Character dialogue games

Strengths

- Completely visual interface
- Quick learning curve

Considerations

Very limited customisation options

tyranobuilder.com

Scratch

Educational programming foundation

Difficulty level:

Beginner ★★☆☆☆

Technical features

- Block-based visual programming with drag-and-drop interface
- Sprite-based character and object system
- Built-in drawing and sound editing tools
- Sharing and remixing platform with global community
- Variables and lists for data management
- Broadcasting system for complex interactions

Best for

- Quiz games
- Mini games
- Simple simulations
- Interactive stories

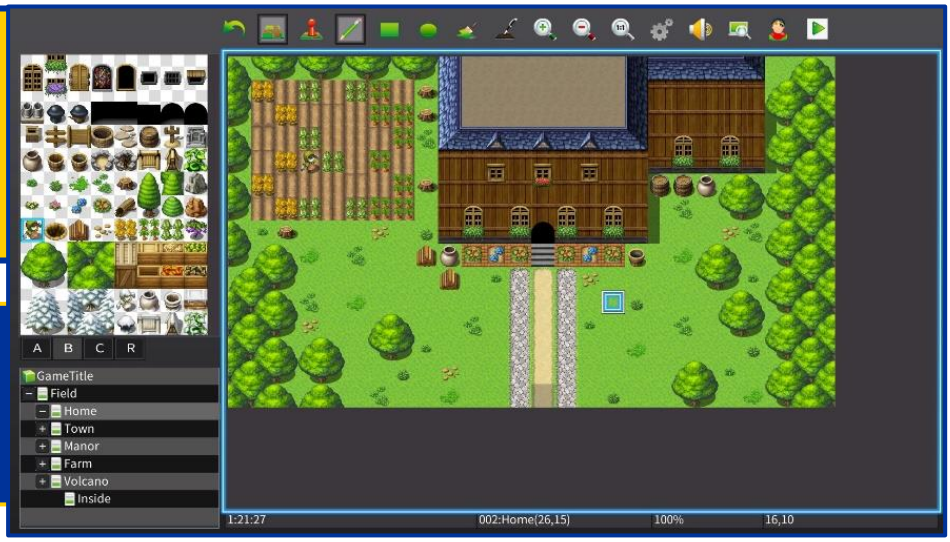
Strengths

- Designed for education
- Extensive curriculum resources
- Completely free
- Minimal hardware requirements

www.scratch.mit.edu

RPG Maker MV

Story-driven adventures



Difficulty level:
Intermediate
★★★☆☆

Technical features

- Event-driven game creation using visual commands
- Built-in character and dialogue systems
- Tile-based map creation tools
- Quest and inventory management systems
- Conditional logic through visual event chains
- Plugin system for extended functionality

Best for

- Historical role-playing adventures
- Character-driven stories
- Quest-based games
- Policy simulations

Strengths

- Powerful storytelling capabilities
- Professional aesthetics possible
- Large community support

Considerations

- Steeper learning curve
- Can be overwhelming for beginners
- Requires time investment

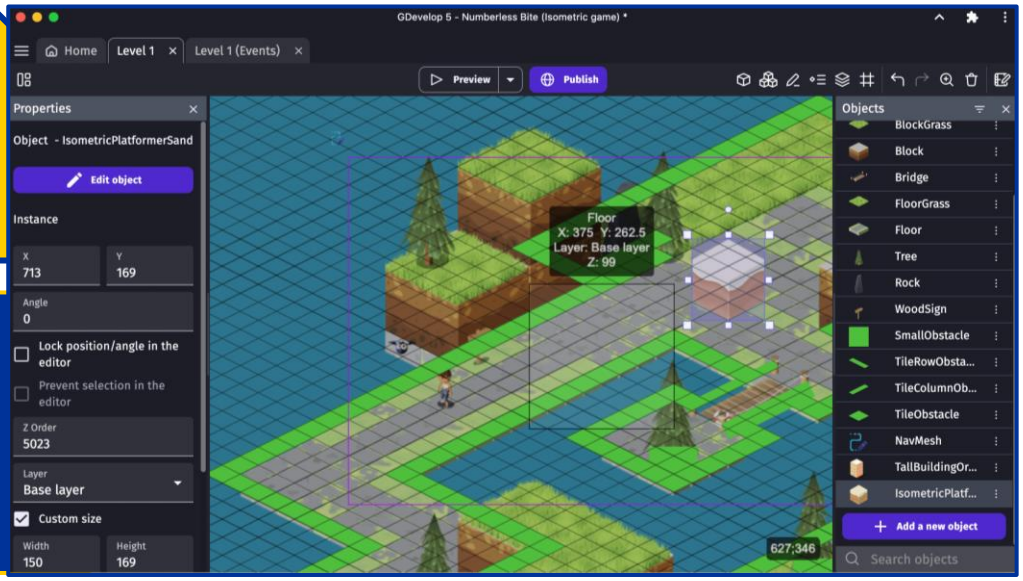
www.rpgmakerweb.com/products/rpg-maker-mv

GDevelop

Visual programming

Difficulty level:
Intermediate to advanced

★★★★☆



Technical features

- Visual scripting system using events and conditions
- Cross-platform publishing (web, mobile, desktop)
- Physics engine for realistic interactions
- Behaviour system for common game mechanics
- Extensive asset library and marketplace
- Real-time preview for immediate testing

Best for

- Interactive simulations
- Physics-based puzzles
- Platform games
- Strategy games

Strengths

- True visual programming environment
- Professional-quality results
- Free and open-source

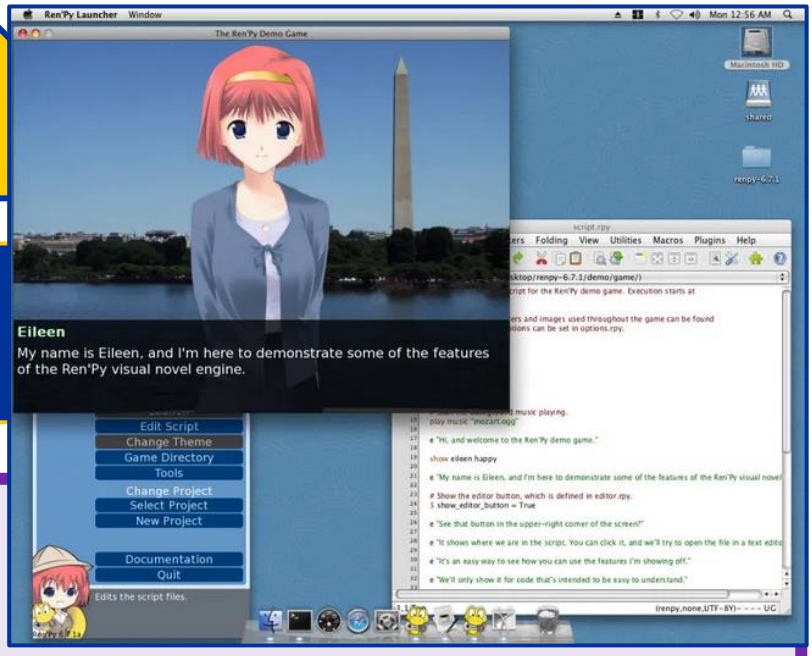
Considerations

- Requires some understanding of programming concepts
- Overwhelming due to the wide range of features
- Longer learning process

www.gdevelop.io

Ren'Py

Visual novel with script



Difficulty level:

Advanced ★★★★★

Technical features

- Multi-language support
- Script-based storytelling with simple syntax
- Character sprite and background management
- Dialogue trees and branching narratives
- Built-in save/load game functionality
- Screen and GUI customisation options

Best for

- Interactive historical narratives
- Ethical dilemma scenarios
- Character-driven stories

Strengths

- Designed for interactive storytelling
- Simple scripting language
- Professional visual novel aesthetics

Considerations

- Requires understanding of programming concepts
- More technical than other options
- Steep learning curve for beginners

www.renpy.org

Chapter 5: Developing multimedia elements and mechanics

5.1 Visual and audio elements: Bringing your game to life

Visual and audio components affect how users interact with and respond to games. They help to **create a specific atmosphere, guide attention, narrative or emotional tone, and support learning** by reinforcing provided concepts. A well-designed scene can **reduce cognitive overload**, as, by definition, it's a situation where one is given too much information at once, or too many simultaneous tasks, resulting in not being able to perform or process the information as it would otherwise happen if the amount were instead sustainable. Cognitive overload in this context means **avoiding too many elements on the screen at once**, so young players don't feel confused, making it easier to understand complex information. Similarly, sound can motivate users, signal progress, or provide feedback during gameplay in a non-direct way.

Even without coding knowledge, beginner creators can enrich their games through various elements. No-code tools like Genially, Scratch, RPG Maker or Construct 3 allow creators to develop these visuals with user-friendly interfaces.

ENHANCING VIDEO GAMES

Characters

Use simple avatars or icons to represent players or roles, helping to establish a clear story and different roles that support and, in a way, narrate the story.

Objects

Include meaningful, theme-related objects that can be clicked, moved, collected or interacted with during the game.

Backgrounds

Choose visual settings that align with the mood, emotion and message of your game.

Simple animations

Add movement or transitions to enhance visual flow or bring attention to key elements, making a story feel more cohesive, intentional and well-shaped.

Besides visuals, sound can dramatically enhance engagement, bringing a game to life, but it should be used carefully. Audio increases engagement by **giving players feedback beyond visuals or written materials**, but it **must not be overwhelming or distracting**. Avoid sounds that feel annoying or unnecessary. There are three main types to consider:

- **Background music**: Sets the tone and mood of the game environment and overall emotion. It should not be too rhythmic or fast, and sound effects should only appear when relevant to the task at hand.
- **Ambient sounds**: Provide contextual realism (e.g., birds chirping in a forest, street noise in a city) and enhance the visuals.

- **Sound effects**: Offer feedback (e.g., a "ding" for a correct answer or a "buzz" for an error), which helps players learn through audio reinforcement.

Many game-making platforms support drag-and-drop audio features, making it easy and not requiring any technical skills.

Tips & tricks

While some game creation platforms include built-in audio libraries, it's smart to have extra sources and tools to expand your sound options. Many online resources offer royalty-free or Creative Commons-licensed sounds:

- ✓ **Freesound**: A vast community-driven database of user-uploaded sounds.
- ✓ **ZapSplat**: Thousands of free sound effects, regularly updated.
- ✓ **SoundBible**: Simple library of free effects ready for download.
- ✓ **Bfxr**: For generating retro-style, 8-bit sound effects.
- ✓ **Audacity**: A free, open-source tool for recording, editing, and mixing audio.
- ✓ **Incompetech**: Free background music composed by Kevin MacLeod, ideal for setting tone or atmosphere in educational games.

5.2 Designing for inclusion and accessibility

Accessibility in digital games is not only about functionality but also about **fairness, inclusion, and creativity**. As video games become a part of daily life through education, entertainment and culture, it is important that games are accessible for everyone. For educators, understanding accessibility is an important step to both **inclusive pedagogy and equitable content design** (Cezarotto, Martinez, & Chamberlin, 2022).

As a step for more inclusive video games, there are a few factors that have to be taken into consideration when designing:

- **Colour palette**: Designers should use high-contrast colour schemes and **avoid relying solely on colour to present information**. This helps users with visual impairments or colour blindness to understand and engage with gameplay

elements better. Tools like colour contrast checkers or filters can assist during development.

- **Text adaptation**: Clear, concise language and readable typography significantly improve the user experience. Important features include **adjustable font sizes, sans-serif fonts, 1,5 line spacing, and left-aligned text**. Subtitles and voice-over options make dialogue accessible to both hearing- and vision-impaired players.
- **Audio experience**: Relying solely on sound for critical game feedback can exclude players with hearing impairments. To address this, developers should include **visual indicators** (like flashing icons or progress bars) or haptic feedback (vibrations or controller responses) for sound-based events for example approaching enemies or successful actions. Additionally, sounds should be calm and not constant. **Avoid continuous or overly stimulating audio**, as it can overwhelm players with sound sensitivity or attention disorders, and may distract from gameplay or learning objectives.

In addition to functionality, game design can support inclusion and representation. **Diverse characters and unique storylines help players to see themselves represented** in gameplay. This not only enhances engagement but also **fosters empathy and social learning**. Representation is not a substitute for accessibility, but when both are present, games become truly inclusive.

Tips & tricks

Educators and developers can follow simplified frameworks such as:

- ✓ **The Game Accessibility Guidelines**: System from basic to advanced, helping to implement features for various impairments.
- ✓ **Microsoft's Inclusive Design Toolkit**: Highlights designing for the benefit of all players.
- ✓ **WCAG Principles** (Perceivable, Operable, Understandable, Robust): While web-focused, many concepts apply to games.

These tools help identify barriers and provide accessible solutions.

5.3 Game mechanics and multimedia logic

Educational game design doesn't demand advanced programming. With visual tools and user-friendly engines, educators can build interactive experiences that support learning goals. By **blending multimedia with simple gameplay**, even beginners can create games that **engage students on cognitive, emotional, and social levels.**

Multimedia elements, such as images, sounds, videos and animations, can enhance the **emotional and sensory dimensions** of a game. When paired with simple interactive mechanics (like clicking, dragging, collecting, or navigating), they foster active participation and let the player have a more immersive experience. For example, dragging a label to the correct image or matching sounds to visual cues can **reinforce learning through multisensory engagement**.

To be effective, these elements should be used with a purpose. Audio should **clarify, not distract** the player. Visuals should **support comprehension, not overwhelm**. Text should be **readable, concise and relevant**. The goal is to enhance user experience without cognitive overload – **affirm one experience and provide quick feedback**.

Modern game creation platforms offer visual scripting systems - based on blocks, switches, triggers, or events - that eliminate the need for traditional coding. These systems allow users to define logic such as: "If the player clicks on an object, then play a sound and reveal text." Beginners can use these systems to define cause-and-effect interactions: **triggering animations, scoring mechanisms, or transitions between scenes**. These mechanics are intuitive and can be directly tied to learning outcomes, for example when identifying correct answers or progressing through knowledge-based challenges.

Tips & tricks

To keep gameplay coherent, youth workers should use **visual planners: flowcharts, storyboards, or level maps**. These tools help structure the sequence of interactions and media. Visual planning clarifies user pathways, choice points, and pacing - whether it's a branching path for decision-making or a linear path for reinforcing knowledge. It also ensures that assets like images, sounds, and feedback align with the learning progression.

As explored in chapter 4, game mechanics shouldn't be random - they should **serve the learning goals**. Matching pairs can support memorisation, simulations can model real-world decisions, and clue-collecting can build critical thinking.

For example: To teach environmental responsibility, a game might have players make choices that affect a virtual ecosystem. Visual, audio and narrative feedback can show the consequences of their decisions, sparking reflection and discussion.

When mechanics are tied to objectives, gameplay becomes more than fun - it becomes **meaningful**. Players learn by doing, testing, seeing, hearing and experiencing, which is the essence of experiential learning.

Mechanic	Multimedia element	Learning outcome	Example
Drag-and-drop	Image and sound	Vocabulary memorisation	Match words to pictures
Multiple choice	Animation and feedback	Critical thinking	Select the correct historical event
Collecting items	Points and visual cues	Environmental awareness	Pick recyclable items to clean a virtual park

Tips & tricks

- ✓ **Give every element a clear job:** Think of each sound, image, or animation as a team member. Treat each sound, image, or animation like a guide. For instance, in a career-planning game, a notification sound can confirm a successful choice, and a brief animation can show progress on a virtual project. Avoid adding background music that makes players anxious or visuals that distract from the task.
- ✓ **Make interaction simple and engaging:** Encourage active participation rather than passive clicking. Example: In a budgeting simulation, players drag and drop money into categories, or match financial strategies to real-life scenarios, reinforcing practical skills.
- ✓ **Align gameplay with learning goals:** Every task should teach something. In a health and wellness game, players could plan a daily schedule with exercise, meals, and sleep, seeing visual and narrative consequences of their choices. In a civic engagement game, collecting clues about local issues can teach research and critical thinking.
- ✓ **Plan the player journey visually:** Map out the game like a mini interactive story. For a sustainability game, sketch which actions affect the virtual environment, when animations or sound effects appear, and what challenges players encounter along the way.
- ✓ **Encourage learning by doing and reflecting:** Let players experiment and see the impact of choices. Example: In a debate or negotiation game, players make decisions and immediately see how it changes the outcome or other characters' reactions, then answer reflection prompts like "What strategy worked best?"
- ✓ **Test, tweak, and iterate:** Run short sessions with youth and gather feedback. For example, in a media literacy game, watch how they identify misinformation; notice what confuses them or what excites them. Adjust visuals, instructions, or interactions based on feedback.

5.4 Playtesting, feedback and learning integration

Educational games reach their full potential only when they are refined through playtesting and user feedback. This process not only **improves gameplay quality** but also **ensures that learning objectives are met effectively**. For educators designing games, **understanding how to test, gather feedback, and iterate based on learner input is vital to creating inclusive, engaging, and meaningful experiences**.

Playtesting involves observing how users interact with the game and identifying what works well or needs improvement. It reveals **usability issues, pacing problems, or confusing mechanics**, but also highlights moments of engagement, creativity, and learning.

In educational contexts, playtesting is also a way to evaluate whether the game supports its intended learning outcomes. It allows educators to **assess if students are engaging with the content meaningfully, making connections, and applying concepts through play** (Salen & Zimmerman, 2004).

When testing games with young players, feedback should be accessible and supportive. Common approaches include:

- **Peer testing**: Learners test each other's games and discuss their impressions informally. This promotes collaboration and ownership.
- **Observation**: Educators observe players during gameplay, noting reactions, confusion, or emotional responses without interrupting.
- **Reflection circles**: After testing, participants can share thoughts in small groups or through simple reflection prompts (e.g., "What did you enjoy?", "What was difficult?", "What would you change?").
- **Visual rating tools**: Emoji charts, traffic lights, or smiley faces can be used for quick, non-verbal feedback collection, especially with younger players.

The aim is to create a **safe space where youth feel their voices are valued** and are not evaluated on right or wrong answers. Effective game design is iterative. Based on feedback, educators can revise games:

- **Multimedia elements:** Replacing confusing sounds or overly complex visuals based on the feed collected from peers.

- **Mechanics:** Simplifying a task, adjusting difficulty – also based on the gathered feedback.
- **Instructions or narratives**: Clarifying goals or improving storytelling flow.

Tips & tricks

Changes should prioritise clarity, accessibility, and alignment with learning objectives. Updates improve user experience but also model creative problem-solving and responsiveness for learners. Educators should reflect on whether the game:

- ✓ Reinforces the intended knowledge or skills.
- ✓ Accommodates diverse learners (e.g., different reading levels, sensory needs, accessibility and inclusivity levels).
- ✓ Encourages positive interaction, cooperation, or empathy.

Assessment doesn't have to be formal; observation and learner feedback can be powerful indicators and happen in non-formal environments. Games become even more meaningful when connected with larger projects or youth work themes. They can support:

- **Self-expression:** Games as storytelling or identity exploration tools.
- **Teamwork:** Cooperation with different users within the game promoting collaboration.
- **Media literacy:** Reflecting on digital design, choices, and consequences.

Linking gameplay to different themes strengthens its relevance and makes learning more interactive. Playtesting is not just a quality assurance step; it is an educational tool. By gathering feedback on real user experiences, educators can refine both the technical and educational aspects of the games.

In conclusion

By following the advice proposed throughout this guide, finding inspiration in the other resources of our project, exploring the no-code engines that fit your needs, developing the mechanics that target your goals, implementing the multimedia elements that enhance your gameplay and following an iterative method to adjust your work, both youth workers and youth are sure to create efficient, engaging and entertaining video games.

With these technical tools, resources and tips, you now have everything you need to dive into video game design, improve youth participation in democratic matters and make civic education more dynamic and interactive.

Bibliography & further readings

Chapter 1: Introduction

- European Commission, Joint Research Centre. (2022). *The Digital Competence Framework 2.2: The Digital Competence Framework for Citizens – With new examples of knowledge, skills and attitudes*. Publications Office of the European Union. https://publications.jrc.ec.europa.eu/repository/bitstream/JRC128415/JRC128415_01.pdf
- Van Audenhove, L., Baelden, D., Mariën, I., & Wuyckens, G. (2024). Data literacy in the new EU DigComp 2.2 framework: How DigComp defines competences on artificial intelligence, internet of things and data. *British Journal of Educational Technology, 55*(2), 555–570. https://doi.org/10.1111/bjet.13447
- Kafai, Y. B. (2016). Constructionist gaming: Understanding the benefits of making games for learning. *Educational Psychologist, 50*(4), 313–334. https://doi.org/10.1080/00461520.2015.1124022
- Berger, S., Kumtepe, E. G., Güntürkün, P., & Slany, W. (2018). Pocket Game Jams: A constructionist approach at schools. *Proceedings of the 2018 IEEE Global Engineering Education Conference (EDUCON)*, 1914–1921. https://arxiv.org/abs/1805.04462
- Lin, H., Chen, Y., & Chang, C. (2024). ScriptAR: No-code development of augmented reality (AR) educational games. *Proceedings of the ACM on Human-Computer Interaction, 8*(CSCW1), Article 64. https://doi.org/10.1145/3734188

Chapter 2: Understanding video game creation

- European Commission. (2022). *Code of practice on disinformation 2022.* Publications Office of the European Union. https://digital-strategy.ec.europa.eu/en/policies/code-practice-disinformation
- European Commission, Joint Research Centre. (2019). *The future of education and skills: Education and training 2020.* Publications Office of the European Union. https://joint-research-centre.ec.europa.eu

- European Schoolnet. (2014). *Games in schools: Teachers' handbook.* European Schoolnet Academy. https://www.europeanschoolnetacademy.eu
- Kenney. (n.d.). *Free game assets.* Kenney. https://kenney.nl/assets
- Ledoux, A. (n.d.). *Bitsy game maker.* itch.io. https://ledoux.itch.io/bitsy
- MIT Media Lab. (n.d.). *Scratch.* Massachusetts Institute of Technology. https://scratch.mit.edu
- OpenGameArt. (n.d.). *Free game art.* OpenGameArt.org. https://opengameart.org
- UNESCO. (2018). *A global framework of reference on digital literacy skills for indicator 4.4.2.* UNESCO. https://unesdoc.unesco.org

Chapter 3: Game development fundamentals

- Block, B. (2020). *The Visual Story: Creating the Visual Structure of Film, TV, and Digital Media*. Routledge. https://doi.org/10.4324/9781315794839
- Cooper, A., Reimann, R., Cronin, D., & Noessel, C. (2014*). About face: The essentials of interaction design (4th ed.)*. John Wiley & Sons. https://www.researchgate.net/publication/318707875_About_Face_The_Essentials_of_Interaction_Design_4th_Edition
- Hackett, M. (2022). *How to make a video game all by yourself: 10 steps, just you and a computer*. Valadria. Kindle edition. https://www.scribd.com/document/846909320/Matt-Hackett-How-to-Make-a-Video-Game-All-By-Yourself-10-steps-just-you-and-a-computer-Valadria-2022-Z-Lib-io
- Schell, J. (2019). *The art of game design: A book of lenses* (3rd ed.). A K Peters/CRC Press. https://doi.org/10.1201/b22101
- Maticz. (2025). *The ultimate guide to game development in 2025*. https://maticz.com/game-development

Chapter 4: Introduction to no-code game engines

- Appmaster. (2023). *Understanding the no-code movement: A historical perspective*. https://appmaster.io/blog/no-code-movement-a-historical-perspective

- Formstack. (n.d.). *History of the no-code movement*. https://resources.formstack.com/reports/rise-of-the-no-code-economy/history
- Games Publisher. (n.d.). *Game development without code: A beginner's guide*. https://gamespublisher.com/game-development-without-coding-a-beginners-guide/
- Games Publisher. (n.d.). *Game development without coding: Which platform is best for you?* https://gamespublisher.com/no-code-game-development-which-platform-is-best-for-you/
- Modd.io. (2023). *No-code game creation: Empowering creativity without coding*. https://www.modd.io/blog/no-code-game-creation-empowering-creativity-without-coding/
- Tonkean. (2023). *No-Code 101: Everything you need to know about the no-code movement*. https://www.tonkean.com/blog/no-code-101-everything-you-need-to-know-about-no-code-movement
- Wired. (2021). *The new startup: No code, no problem*. https://www.wired.com/story/new-startup-no-code-no-problem

Engine selection and description:

- CodeOrNoCode. (2023). *The best no-code game engines for game development*. https://codeornocode.com/no-code/best-no-code-game-engines
- CreativeBloq. (2023). *The best no-code game engines recommended by leading indie devs*. https://www.creativebloq.com/3d/video-game-design/leading-indie-devs-recommend-their-favourite-no-code-game-engines
- Directual. (2023). *The best no-code game engines in 2023: Coding-free game development*. https://www.directual.com/blog/no-code-game-engines
- Dzun_n. (2020). *These 5 game engines don't require coding to make games*. https://dev.to/dzun_n/these-5-game-engines-dont-require-coding-to-make-games-5do1
- GameDevLab. (2023). *The best no-code game engines: Create games without coding*. https://www.gamedevlab.net/en/post/the-best-no-code-game-engines-create-games-without-coding
- Genies.com. (2023). *Best no-code game makers for beginners: A comprehensive guide*. https://genies.com/blog/best-no-code-game-makers-for-beginners-a-comprehensive-guide

- Hyperpad. (2023). *Top no-code game creation tools for beginners*. https://www.hyperpad.com/blog/top-no-code-game-creation-tools-for-beginners
- IndieGameCloud. (2023). *Game engines that don't require coding*. https://indiegamecloud.com/game-engines-that-dont-require-coding/
- PeerDh. (2023). *Game development: Essential software for every developer*. https://peerdh.com/blogs/programming-insights/game-development-essential-software-for-every-developer
- SHNO. (2025). *Best no-code game engines - 2025 review*. https://www.shno.co/blog/no-code-game-engines

Chapter 5: Developing multimedia elements and mechanics

- Bers, M. U. (2020). *Coding as a playground: Programming and computational thinking in the early childhood classroom (2nd ed.)*. Routledge. https://www.taylorfrancis.com/books/mono/10.4324/9781003022602/coding-playground-marina-umaschi-bers
- Cezarotto, M., Martinez, P. N., & Chamberlin, B. (2022). *Developing inclusive games: Design frameworks for accessibility and diversity*. In *Game Theory – From Idea to Practice*. https://doi.org/10.5772/intechopen.108456
- Gee, J. P. (2003). *What video games have to teach us about learning and literacy*. Palgrave Macmillan. https://www.researchgate.net/publication/220686314_What_Video_Games_Have_to_Teach_Us_About_Learning_and_Literacy
- Mayer, R. E. (2001). *Multimedia learning.* Cambridge University Press. https://assets.cambridge.org/97805217/35353/frontmatter/9780521735353_frontmatter.pdf
- Plass, J. L., Homer, B. D., & Kinzer, C. K. (2015). Foundations of game-based learning. *Educational Psychologist, 50(4),* 258–283. https://doi.org/10.1080/00461520.2015.1122533
- Salen, K., & Zimmerman, E. (2004). *Rules of play: Game design fundamentals*. MIT Press. http://files.onearmedman.com/fordham/salen_zimmerman2004.pdf
- Statista. (2024). *Share of internet users who played video games in Q3 2024, by age group*. https://www.statista.com/

Annexes

Additional video game engines (all levels)

Bitsy

www.bitsy.org

Technical features

- Ultra-simple tool for creating tiny story-driven, pixel-art games.
- No audio, minimal animation, text-based interactions.

Best for

- Games focused on dialogue, empathy, civic dilemmas.
- Storytelling, reflective or narrative experiences.

CONSTRUCT 3

www.construct.net

Technical features

- Fully no-code engine using "event sheets" (if/then logic).
- Works directly in a browser; no installation needed.
- Great for responsive UI and educational interactions.

Best for

- Simulations, storytelling with choices, quizzes, management games.
- Professional results without coding and with limited hardware access.

stencyl

www.stencyl.com

Technical features

- Drag-and-drop, block-based interface.
- Export to desktop, mobile, HTML5.
- Optional advanced scripting for growth.

Best for

- Adventure, puzzle, or platformer games with limited to no coding.
- Quick prototyping with accessible logic blocks.

MINECRAFT EDUCATION

education.minecraft.net

Technical features

- Sandbox world-building game used widely in schools.
- Supports collaborative multiplayer worlds.
- Built-in tools for lessons, quests, NPCs, coding challenges.

Best for

- Cooperative civic missions: community building, resource management.
- Teaching democratic decision-making through co-created worlds.

GameMaker™

gamemaker.io

Technical features

- Powerful 2D engine with a visual interface and its own scripting language.
- Excellent for professional-quality 2D games (platformers, puzzles, RPG).
- Large community, many tutorials, asset marketplace.

Best for

- Polished 2D gameplay, animations, and complex mechanics.
- Projects where participants already have some coding experience.

ROBLOX Creator Hub

create.roblox.com

Technical features

- Massive creation platform where players build multiplayer games.
- Uses Lua for scripting mechanics, items, behaviours, or multiplayer systems.
- Huge youth player base, strong community tutorials.

Best for

- Multiplayer civic games (e.g., shared governance, town building).
- Long-term youth clubs or advanced workshops.
- Projects exploring online communities, moderation, digital citizenship.

UNREAL ENGINE

www.unrealengine.com

Technical features

- Industry-standard engine for AAA-quality 3D games, professional results.
- Blueprints system without typing code, but complexity remains high.
- Requires strong digital skills and significant time investment.

Best for

- Realistic simulations and immersive environments.
- 3D graphics, VR, museum experiences, city builders.

EU VIDEO GAMES

www.eu-videogames.eu

Co-funded by
the European Union

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the Europear Union or the European Education and Culture Executive Agency (EACEA). Neither the European Union nor EACEA can be held responsible for them.

Project code: 2024-2-PL01-KA220-YOU-000286883

Akademia Humanistyczno Ekonomiczna w Łodzi
EGLE GAMIFICATION IN LEARNING AND EDUCATION
euphoria Net
TAVO EUROPA
sempre a frente