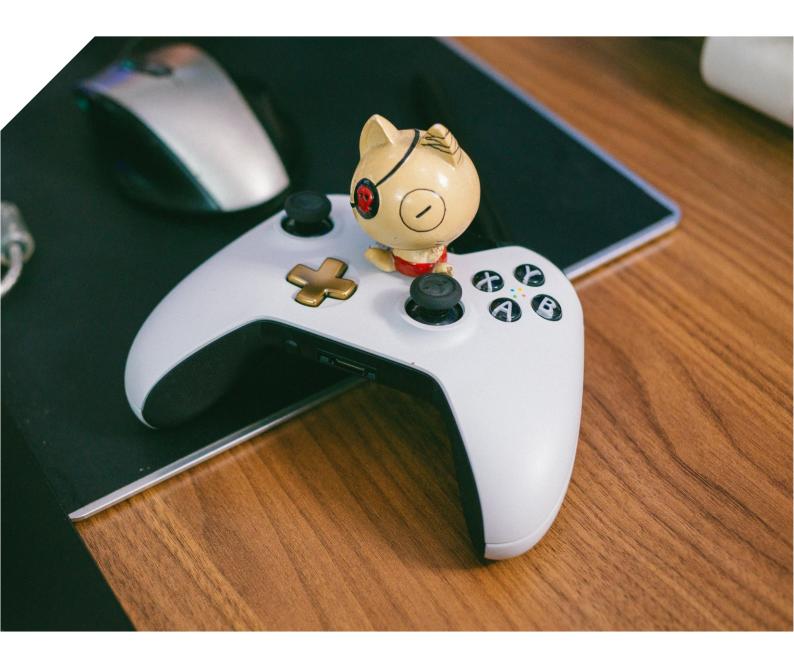LABORATORIES ACROSS BORDERS 4 ALL

Basics of the **Gamification** concept and its potential for the subject **Control Engineering**

- 2 -

Citation of the work:

GEFÖRDERT VOM

Bundesministerium
für Bildung
und Forschung

Executive Summary

The Research provides a comprehensive overview of the online education industry for technical students. It covers the current state and growth trends of online technical education, along with a detailed analysis of the target audience, including demographic, psychographic, and behavioral factors. The report also assesses the educational needs and motivations of students, and examines the perspectives of various stakeholders such as students, parents, education platforms, and employers.

Furthermore, the report explores worldwide trends, the evolving challenges faced in technical education, and provides projections on the sector's future direction.

The report's key contribution is mapping out crucial skills and competencies for engineering students specializing in areas like electric power engineering, electromechanics, industrial automation, mechatronics, and robotics. It uniquely connects these identified skills to specific online educational platforms, offering a practical approach for enhancing these skill sets. This feature makes the report a useful resource for guiding students and educators towards targeted online resources for skill development in these technical fields.

Disclaimer

This work summarizes web-available content, as cited and explained under the section "Literature". This report is intended to provide an overview without taking credit for the concepts and definitions contained herein. The authors of this document do not claim any copyright on the work presented herein.

Acknowledgements

Contents

Introduction

The number of students has been declining in the recent years, especially in technical majors. Among other things, the loss of interest in such courses should be one of the main reasons of this problem.

Trying to change the old school methods of teaching depending on modernization and being up to date with teenagers' interests should be the right answer of this problem.

To modernize teaching methods, Reutlingen University would like to introduce and try out new concepts in the learning process, which their effect and impact would be observed, measured, analyzed, and rated.

One promising concept is learning by playing. This involves the learner acquiring certain learning material in the form of or with the help of a game. This project is an attempt to put this concept into practice.

Project goals

The main goal of this project is to develop a game using the Gamification concept which helps master and bachelor students of the Faculty of Engineering at Reutlingen University to review the basics of control engineering.

Through the interaction between students and the game, independent learning should take place. By the end of the game, the basics of control engineering should be internalized.

Gamification

Gamification is adding game mechanics into nongame environments, like a website, online community, learning management system or business' intranet to increase participation. The goal of gamification is to engage with consumers, employees, and partners to inspire collaborate, share, and interact.

The two main types of Gamification

The two gamification types, content and structural, function differently, but both still provide advantages to learning

Structural gamification

Structural gamification involves applying game elements to move learners through content with no changes or alterations to the content itself. This type of gamification focuses on motivating employees to work through the content, keeping them engaged by offering rewards.

One example of this type of gamification in training is allowing staff to earn points for watching an instructional video or completing a training activity. The points incentivize training completion as employees work to gain the most points they can.

It's important to remember that structural gamification adds external factors to make an activity game-like but doesn't change the content. Even if story elements, like characters, are a part of gamification, the learning content remains the same. Content gamification

Content gamification

In the case of content gamification, the training content itself is altered to make it more game-like. It doesn't necessarily have to become a full serious game but instead adds games or activities into the content. The focus is to increase user engagement by attaching interactive elements like challenges, feedback loops, storytelling, and learning from mistakes to offer an engaging training experience.

In this type of gamification, employees explore and learn material without being provided learning objectives from the beginning. For instance, staff could explore a virtual tour and locate hidden items, indicating an understanding of the training topic through the game mechanics within the training. The tour is part of the training content itself.

Another example of content gamification is beginning the training with an entertaining activity to introduce the course and intrigue users instead of starting with a list of learning objectives and getting rewarded after completing specific tasks.

Relevance for control engineering

The type Structural Gamification is more relevant here, Since the focus is on the content respectively the basis of control engineering in this project.

- 5 -

Elements of gamification[1]

There are several elements of Gamification, which would be listed in the following table. Most certain some of these elements are going to be used in this project but not all of them.

Game Element	Examples	Usefulness to Instructional Design
Achievement (Progression)	» Points » Badges » Leveling » Leaderboards » Progression bars » Certificates	Game players get satisfaction from level accomplishment and skill development. Learners enjoy the same types of recognition. The sense of progression motivates continued effort. Leaderboards provide a social status element, as do points and badges. In training, the course completion certificate signals achievement.
Rewards	» Equipment, tools, and other resources to use in game » Collectibles » Bonuses » Power-ups	Closely related to achievement, rewards can be scheduled into the learning experience. Both variable and fixed reward schedules are popular game mechanics. Rewards can be based on completing several actions or distributed at set intervals. Rewards provide extrinsic motivation and recognition for time, effort, and skills attained.
Story	» Narrative arc » Quest: The hero's journey	An adventure setting, a thwarting disaster scenario, or a beating the competition narrative pique learner interest and motivation. Put the learning experience into a compelling narrative setting. Add characters, conflicts, and resolution to immerse the learner – and learner choices – into the storyline.
Time	» Countdown » Schedule	A common trope in board games, timers (counting total time) and countdown clocks create a sense of urgency. Even using a schedule of events, e.g., before I do B and C, I must complete A, helps focus learner attention to the task at hand.

---

[1] As presented by Mindy Jackson [1]

Game Element	Examples	Usefulness to Instructional Design
Personalisation	❯❯ Avatar selection ❯❯ Avatar customization ❯❯ Character naming Interactive conversation (ICI)	With HTML5 and dynamic CSS, it's easier than ever to provide personalization. From selecting and customizing an avatar to choosing the look-and-feel options (e.g., a dreamy theme or a bright color theme), accommodate individual preferences. Use the information from learner input fields. For example, if the learner inputs a nickname into a text field, use that nickname within the environment or narrative. Repurpose previous responses to provide a sense of intelligence or awareness: "I am known." For example, pull up an earlier answer response later in the game. Personalization ups learner engagement and motivation.
Micro-interactions	❯❯ SFX ❯❯ Toggles ❯❯ Animated rollovers Easter eggs	Details matter when creating a great experience. Games provide numerous and satisfying moments and micro interactions: a hover-state animation, a sound effect, or a cut-screen narration. But beware of too much flare! Provide nuanced environmental reactions to learner actions through sound, subtle animations, and cool transition screens.

Potential approaches

At the beginning of the project it is important to make some decisions about the characteristics of the game. For example, how should the game look like or how should it be programmed.

Game view

It's important to decide which game view is more suitable for this project 2D or 3D. Programming, Art, and Design are the three keywords for this comparison.

3D is much more charming than 2D games, but this comes with a high coast and difficulty grade which going to be explained here.

Programming

❯❯ The math is significantly more complex for rendering, physics, collision, etc. (matrices and vectors).

❯❯ Because of the previous point, good performance is much more difficult to attain. With today's hardware, one can make a nice-looking 2D game without having to think about performance at. With 3D, one will have to do some optimization.

- » The data structures are much more complex. Because of the previous point, one will need to think about culling, space partitioning, etc. all of which are more challenging than a simple "here's a list of everything in the level".

- » Animation is much more complicated. Animation in 2D is just a filmstrip of frames with possibly different positions for each frame. With 3D, one will need to deal with separate animation assets, bones, skinning, etc.

- » The volume of data is much higher. One will have to do intelligent resource management. Games come with gigabytes of content, but consoles certainly don't come with gigabytes of storage.

- » The pipelines are more complex to develop and maintain. One will need code to get assets into the engine's preferred format. That code doesn't write itself.

Art

- » The assets are, of course, much more complex. One will need textures, models, rigs/skeletons, animation, etc. The tools are much more complex and expensive, and the skills to use them harder to find.

- » The set of skills needed is wider. Good animators aren't often good texture artists. Good lighters may not be good riggers.

- » The dependencies between the assets are more complex. With 2D, one can partition any assets across different artists cleanly: this guy does level one, this guy does enemies, etc. With 3D, the animation affects the rig which affects the skeleton which affects the model which affects the textures which affect the lighting... The art team will have to coordinate carefully and constantly.

- » The technical limitations are more complex to deal with. With 2D it's basically "here's your palette and your max sprite size". With 3D, the artists will have to balance texture size (for multiple textures: specular, color, normal, etc.), polygon count, keyframe count, bone count, etc. The particulars of the engine will place random weird requirements on them.

- » Asset processing takes longer. Pipelines to convert 3D assets to game-ready format are complex, slow, and often buggy. This makes it take much longer for artists to see their changes in game, which slows them down.

Design

- » User input is complicated. One must deal with camera tracking, converting user input into the character's space intuitively, projecting 2D selections into world space, etc.

- » Levels are hard to author. The level designers basically need the skills of a game designer and an architect. They must consider players getting lost, visibility, etc. when building levels.

- » Level physics is tedious to author. One will have to check and recheck and recheck again to make sure there aren't gaps and bugs in the level physics where players can get stuck or fall through the world.

» Tools are much harder. Most games need their own tools for authoring things like levels. Since the content is so much more complex, the tools are more work to create. That usually results in tools that are buggier, incomplete, and harder to use.

Summarized, a 3D game would definitely leave a better impression by the players, but the coast of developing time, complexity and other resources is too high for a gamification project with the purpose of teaching technical knowledge. That has been said, a 2D game would be more suitable option for this project, considering the capabilities of the staff and what it is to be achieved.

Game Engine[2]

There are plenty available game engines out there, each with his its own characteristic, advantages, and disadvantages. In the following some of the best of them would be discussed. Please be advised, that this review may change over time and it represents the current state as of 2023.

Unity

Game developers have relied on the Unity gaming engine since 2005. With continuous updates and new significant features added every year, such as Unity Reflect, the engine enjoys an unbelievable level of support. With many companies and developers creating convenient SDKs for the machine, it is not only a popular choice for 2D and 3D games but also virtual reality and augmented reality.

Advantages of unity game engine:

» For beginners with incomes under $100K, it's free

» Suitable for 2D and 3D games

» Support for the development of mobile games

» SDKs for VR and AR are available

» Free assets at the Asset Store

A few weaknesses of Unity:

» Professional licenses are expensive

» Computers with better performance are required for higher-end tech demos

» Changes to the UI

Unreal

Today, many of the most popular triple-A games are powered by Unreal Engine, a powerful graphical engine featuring lighting, shaders, and more. The unreal engine handles several complicated tasks

---

[2] Based on Joseph Sibony [5] and perforce.com [6]

more efficiently due to its wide use in the game development sector. Also, the community is constantly improving the unreal engine.

The visual blueprinting feature of the unreal engine allows even non-programmers to design games, and it's an all-around powerhouse that takes VR to new heights.

It has been reported that Unreal Engine is a better fit for large projects and projects that one plans to work on as a team. The program also requires a more powerful computer due to its heavy graphics requirement compared to other engines like Unity.

Unreal Engine is not necessarily the best engine for creating 2D games, even though it can make 3D games. Students worldwide have created some fantastic things with Unreal Engine features. It might be interesting to know the difference between Unreal Engine vs. Unity 3D.

Advantages of unreal engine:

>> Graphics enthusiasts will love it

>> The engine that performs better than others

>> VR's top pick

>> Non-programmers can use visual blueprints

>> Unrestricted assets are available on a large marketplace

Disadvantages of unreal engine:

>> Simple, or solo projects are not recommended

>> Computers with high-performance graphics are required

>> Gameplay is better in 3D than in 2D

Godot

Despite Godot's game development existence since 2014, the engine has recently become increasingly popular. If one is looking for something less complex and open-source, one can use the Godot game engine and develop games on his own way.

In addition to supporting both 2D and 3D graphics, the engine is well-suited for making any kind of game.

Furthermore, Godot game development takes a unique approach to representing specific game functions with its node and scene architecture, making it stand out from competitors and more user-friendly. One has got a real winner when pair it with a video game development company.

The scripting language Godot uses is called GDScript, which has some weaknesses. Since the language was designed specifically for Godot game development, it is like Python (a favorite language for many

developers). Still, experienced game developers may find it tedious to learn yet another language. As Godot is less known than a prominent engine like Unity or Unreal, fewer resources are available.

Advantages of Godot Game Engine:

>> Games in 2D and 3D are supported

>> Open-source and free to use – even commercially

>> Community of passionate individuals

>> Developing games with a unique architecture

Disadvantages of Godot Game Engine:

>> GDScript may not appeal to experienced game developers

>> Compared to other engines, it has fewer resources

Phaser

Phaser was launched initially in 2013, but the most recent version, Phaser 3, was released in 2018. Despite this, developers who want to develop mobile or browser-based games continue to find the framework extremely popular.

Phaser is primarily based on web development technologies and is a relatively stable framework even as new versions are released. Besides this, Phaser offers physics and all the other kinds of features one would expect from an engine, so it's possible to make any game with it, including MMORPGs.

It's important to note that Phaser game development relies on 2D graphics, which means 3D game aren't makeable with it. While phaser's framework is cross-platform, some limitations persist that cannot be found in more powerful, stand-alone engines.

Advantages Of Phaser Game Engine:

>> Playable in browsers and on mobile devices

>> The system is highly stable

>> With web technologies, game development can be easily expanded

Disadvantages:

>> Specially designed for 2D games

>> Engines with fewer capabilities

GameMaker Studio 2

The popular GameMaker Studio game engine created in 2017 is the newest version of a product that has existed since 1999 under many names and iterations. Games developed using the GameMaker game engine are supported on a wide range of platforms, including Nintendo Switch.

In addition to a visual editor, GameMaker Language can be used to program custom behaviors that go beyond which is available with visual programming. As an easy-to-use game engine, everyone can easily develop games with it.

As a proprietary game engine and framework, GameMaker Studio 2 is not ideal for those looking for a budget-friendly solution. Aside from that, it is also geared explicitly towards other engines like Phasers. The game engine developed does have some limitations in terms of 3D capabilities, but it cannot compare to the visual editor of Unity, Unreal, or Godot gaming engines. It might be interesting to find list of best gaming name generator.

Advantages GameMaker Studio 2:

>> Numerous platforms are supported

>> Programming is easy with drag-and-drop

>> Easy to use for beginners

A few weaknesses:

>> Designed primarily for 2D games

>> Obtains for a price

Conception

After learning about the project goals, requirements, and possible approaches its time to put a draft of a project concept. For this, a potential development path through the various possibilities must be followed with the help of some evaluation criteria.

Engine selection

The following evaluation table should help finding the suitable engine for this project with the help of the project requirements as criteria and few other game engine relevant criteria.

The highest Priority is one and the lowest is four. The Game Engine with the least rating points respectively the lowest ranking is the most suitable Engine for this project.

Criterion	Priority	Unity	Unreal	Godot	Phaser	GMS2
Platform	1	1	1	1	1	1
CPU friendly	1	2	2	1	1	1
Programming friendly	2	1	1	1	1	1
After service	1	1	1	1	1	1
Cost effective	2	3	3	1	1	3
Offline mode	3	1	1	1	1	1

Criterion	Priority	Unity	Unreal	Godot	Phaser	GMS2
Open source / community	2	1	1	1	2	1
Engine capabilities	4	1	1	2	3	1
Points		11	11	8	11	10
Rating	$\sum Prio * \frac{1}{Points}$	7,083	7,083	**5,33**	6,083	6,083
Rank		3	3	**1**	2	2

According to the results of the rating, the game engine Godot is the most suitable engine and the one to use in this project. For this purpose, an online tutorial is being conducted.

Conclusion

This study presented the basic elements of gamification, which can be integrated into the topic of control engineering. Due to Europe's strict regulations on Data Protection (EU-GDPA) it is suggested to use only the elements which can engage the student and spark the intrinsic learning drive and to let elements like competitions, collectibles and personalization (like avatars) for later development stages.

As for the implementation, it is suggested to use an open-source game engine in a 2-D environment as an entry point. The preferred deployment method would be a web-based application with low requirements on the end devices.

Literature and Sources

[1]   Jackson, M., **Gamification Elements to Use for Learning**, *Presentation, available online*. 2016 Enspire.

[2]   Lee, Joey & Hammer, Jessica. (2011). **Gamification in Education: What, How, Why Bother?**. Academic Exchange Quarterly. 15. 1-5. Available online: https://www.researchgate.net/publication/258697764

[3]   Barney, N., **Gamification Definition**, from techtarget.com, Available online: https://www.techtarget.com/searchhrsoftware/definition/gamification

[4]   Selawsky, J., **Gamification In Education: A New Type Of Interactive Learning**, Blog post on 18/9/2019, from elearningindustry.com, Available online: https://elearningindustry.com/gamification-in-education-new-type-interactive-learning

[5]   Sibony, J., **The Best Game Engines You Should Consider for 2023**, Blog post on 2/2/2023, from incredibuild.com. Available online: https://www.incredibuild.com/blog/top-gaming-engines-you-should-consider

[6]   perforce.com Blog, What Are the Best Game Engines?, available online: https://www.perforce.com/blog/vcs/most-popular-game-engines