Originally published July 2022 in
Journal of Education and Learning; Vol. 11, No. 5; 2022
ISSN 1927-5250 E-ISSN 1927-5269
by Canadian Center of Science and Education
https://doi.org/10.5539/jel.v11n5p183

A Survey of Estonian Video Game Industry Needs

Raimond-Hendrik Tunnel[1] & Ulrich Norbisrath[1]

[1] Institute of Computer Science, University of Tartu, Tartu, Estonia

Correspondence: Raimond-Hendrik Tunnel, Institute of Computer Science, University of Tartu, Tartu, Estonia, Narva mnt 18, 51009, Tartu, Estonia.

Abstract

Designing a video game design and development curriculum in higher education is a challenging task. Information about the needs of the respective industry certainly helps. In this paper, we have surveyed Estonian video game development companies to determine their current needs when it comes to knowledge areas, software tools, languages, abilities, and contextual fluencies. The survey is based on a similar survey conducted a decade ago and this paper compares the current results with those found earlier.

Compared to the prior survey, we have found significant differences in the rated importance of knowledge in optimization, version control technologies, the C, C++, and C# programming languages, and the time management ability for video game development companies looking to hire university graduates. We have also extended the previous survey to include a contemporary selection of game design and development tools. Based on that, we have determined a strong need for graduates with skills specifically in Unity and Unreal Engine game engines, Photoshop raster image editing software, and Git version control software.

While most of our results are largely consistent with the previous research, our added survey items like visual languages and game engines bring the results to the modern context. This allows curriculum designers and managers to see the differences regarding the landscape of industry needs for their graduates and thus make more informed decisions in their work.

Keywords: Contextual fluency, curriculum design, game design tools, game development tools, game engines, higher education, programming languages, skills, video games, video game industry.

1. Introduction

Over the years, the video game industry and video game curricula in higher education institutions have become increasingly common. Popular game engines such as Unity and Unreal Engine make video game development available for everyone with sufficient programming skills. There is very good, established literature as well as practices for designing video game user experiences (Rogers, 2014; Salen & Zimmerman, 2003; Schell, 2008; Swink, 2008). Publicly available recordings of talks from conferences such as the Game Developers Conference make even the bleeding edge techniques available for everyone.

While readily accessible, truly learning and mastering video game design and development takes time and effort. Higher education institutions offer many Bachelor's and Master's level curricula in different aspects of video games (Valentine, 2014), ranging from programming and development, media and design, artistry, marketing, business, and entrepreneurship, to game studies and analysis (Bouchrika, 2021). No matter the path, these fields are not easy and require dedication from the students.

Thus, when designing such a curriculum, it is essential that the skills and knowledge it provides form a useful set for the graduates. They need to be well prepared to continue their studies at the next level and find junior-position employment in the industry. Balancing or focusing the curriculum among these complexions of the video games field can be quite challenging for the curriculum designer (Mateas, 2007).

At the University of Tartu, we are deliberating on a new video game designer-developer Bachelor's level curriculum. We have chosen specializations in video game design and development as these are of primary importance in making games. Some of our current computer science curriculum courses can be used in the development specialization. However, including the teaching of video game design is something we feel very passionate about as a video game with mediocre design but excellent implementation would still provide just a mediocre experience.

Our experience in teaching video games comes from running elective courses and conducting thesis supervision in our existing computer science and software engineering curricula. It has become obvious that students in such general studies make a lot of extra effort in their final year to prepare themselves for working in the video games field. And even doing so will, at times, still leave them inadequate in the eyes of the respective companies.

When designing our new video games curriculum, we want to make sure that the three years of study will be efficient for both students and educators. To make informed decisions regarding curriculum design, we need to know what the industry expects from our graduates. As the video games industry in Estonia is relatively young, it is unclear if the learning outcomes and practices from countries with well-established industries would work here or not. As such, we surveyed the video games, computer graphics, and virtual and augmented reality companies to map out their expectations.

2. Background

There are many success stories of curriculum design in different ecosystems and with different purposes. For example, Mikami et al. (2010) describe their very practical 4-year curriculum developed and run at the Tokyo University of Technology. Fachada and Códices (2020) propose a top-down interdisciplinary curriculum for the Portuguese private university Universidade Lusófona. Kenwright (2016) has developed a *holistic* curriculum, where the program provides a wholesome learning experience, giving both a broader understanding and the essentials required in the field.

These examples of innovation in curriculum design certainly provide insight and offer options to mix and match in designing a new curriculum in another ecosystem. However, they do not describe a general baseline upon which to build it. This becomes apparent even in the details, for instance, what tools or languages are being taught. Kenwright (2016) mentions that they do not focus on one language so as to give the students better problem-solving and learning skills. They mention, as examples, the Python, Java, C++, and HTML languages. However, it is unclear if these are the languages also valued by the industry and, if so, how much. A similar example can be found in the paper about the curriculum in Tokyo (Mikami et al., 2010), where they describe using Maya when teaching students to create 3D graphical assets. This begs the question if Maya is the best choice for the purpose. Do the industries value the graduates' skills with Maya, would they prefer that they knew another tool, like Blender, instead, or does this particular choice matter to them?

An extensively referenced document for the baseline of a video games curriculum is the IGDA Curriculum Framework from 2008, developed by the International Game Developers Association (IGDA, 2008). The framework, albeit more than a decade old now, lists nine different core topics with many sub-topics under each. Fields range from all the different aspects of video games mentioned earlier (game studies, development, design, business, etc). The document is extensive, but there is no ranking of the different sub-topics. For example, databases and machine architecture are on par with software design patterns and game logic scripting languages. It is unclear how much focus these fields should have in a video games curriculum.

For a better overview, McGill (2009) has made a thorough survey and comparison of both industry needs and existing emphases in curricula in higher education institutions. They had conducted content analysis on advertisements for game developer positions (McGill, 2008). From that they developed a survey, which consisted of five categories: 1) *Knowledge Areas*, 2) *Languages*, 3) *Software Tools / Environments*, 4) *Abilities*, and 5) *Contextual Fluencies*. They then had game development companies and higher education institutions with a game developer field in the US and Canada rate different items in these categories on a 5-point Lickert scale. This comprehensive survey not only allowed to identify mismatches between the industry needs and curriculum focus but also to acquire informative overviews of both individually.

The downside of the survey by McGill is that it is also more than a decade old now and was done before the popularization of game engines (Andrade, 2015). One decade is sufficient time for the landscape of a new field like video game development to have changed considerably. Furthermore, McGill surveyed the companies in the regions of the US and Canada where game development was an established field. It is possible that in ecosystems where the field is still in the starting phase, the needs of the industry will be considerably different. Thus, a comparable survey made in Estonia seemed like an important and potentially very informative first step in designing a new curriculum.

3. Method

We created a questionnaire to collect the data and analyze the needs of Estonian video game, computer graphics, and virtual reality development companies. The questionnaire was based on the survey conducted by McGill (2009) with some of the response options modernized. We also asked about specific tools used in these companies and to what extent the skills in these specific tools are necessary for their applicants.

3.1 The Questionnaire

Similarly to the previous study, we asked about *Knowledge Areas*, *Languages*, *Software Tools / Environments*, *Abilities*, and *Contextual Fluency*. We mostly used the same items for each category as McGill, which were established in their previous paper (McGill, 2008) based on real-world job applications. The questionnaire asked to rate the importance of each item on the Lickert scale from 1 (*Strongly disagree*) to 5 (*Strongly agree*). As such, direct item-wise comparisons of the results could be made. The questionnaire was in English.

However, during pre-interviews with university colleagues and industry practitioners, it became obvious that some of the items from the 2009 survey had become outdated. Furthermore, the ecosystem of different software tools and environments has grown considerably. Thus, we made a few changes to the original item lists. Naturally, the added or removed items would no longer allow for individual comparisons for these specific items. Most items did remain unaffected and thus most results do remain comparable. The changes are as follows:

- Added Software Testing, Video Game Analysis, and Video Game Design to the *Knowledge Areas* category.
- Added GLSL / HLSL, Go, JSON, Rust, TypeScript, and Visual Languages to the *Languages* category.
- Changed the HTML/DHTML/XHTML item to just HTML.
- Removed OpenGL from the *Languages* category because it is not a language.
- Removed ActionScript from the *Languages* category because the Flash technology from Macromedia (later Adobe), which mainly used that language, had become obsolete in the interim.
- Removed Ceramics from *the Contextual Fluency* category as it was found the least important in the original survey, and we deemed it unnecessary in this context.

The exception is the *Software Tools / Environments* category. Based on the input from pre-interviews, that category was completely remade. Instead of individual tools, we opted for the following more general items:

- 3D content creation tool (Blender, Maya, Houdini, 3DS Max, Modo)
- Audio editor (Audacity, ...)
- Database server (SQL, NoSQL)
- Digital audio workstation (Ableton Live, Audition, FL Studio, REAPER)
- Game engine (Unity, Unreal Engine, Godot, GameMaker Studio, Open 3D Engine, CryEngine, Construct 3, Source 2)
- Graphics API (DirectX, OpenGL, Vulkan, WebGL)
- IDE (Visual Studio, Visual Studio Code, Monodevelop, Rider)
- Project management tools (Confluence, Jira, Trello)
- Raster graphics editor (Photoshop, PaintShop Pro, Affinity Photo, Krita, GIMP)
- Software runtime (.NET Framework / Mono, Java, Node.js)
- Texturing tools (Substance 3D, Material Maker, ArmorPaint, Quixel Mixer)
- Vector graphics editor (Affinity Designer, Adobe Illustrator, Inkscape)
- Version control technology (Perforce, Git, SVN, Mercurial, Unity Collab, Plastic SCM, Dropbox, Google Drive)
- Video editor (DaVinci Resolve, Adobe Premiere, OpenShot, Kdenlive)
- Visual effects tools (After Effects, Nuke, DaVinci Resolve).

The questionnaire proceeded with individual questions about all the specific tools previously listed in brackets for each item (except for the *Database server*). In these questions, the participants were asked to mark for each category the tools used in the company. Next to each question, there was a secondary question “How important are the skills in these specific tools for a candidate?” with the following options:

- *It is OK if they do not have skills in these or alternative tools.*
- *It is OK if they have skills in alternative tools.*
- *They need to have skills in these specific tools.*

The answers to these questions would allow us to home in on what specific tools are required in the given companies and, thus, what should be used when teaching the curriculum. Among the last questions, we asked how many new graduates the company would be willing to hire every year, assuming they meet their recruitment criteria. That number would allow us to quantify the previous answers and make sure that we include in our teaching the technologies in which skills are needed in the industry.

3.2 The Participants

The participants for the survey were found from six different sources. The first source was the website shipped.ee (Anonymous, 2022), which is a privately managed website listing released Estonian PC games and their developer companies. From the list, we filtered out companies that had released a video game during the last 10 years. That strategy excluded companies that had only been active in game development by releasing a relevant product more than a decade ago and thus most likely were irrelevant for the current job market. The second source was a members list of the Estonian Virtual and Augmented Reality Association (EEVR, 2022). From that list, we looked for private companies that develop or publish virtual reality software themselves. This excluded labs from universities, museums, and government institutions. The third source was a list of gaming startup companies in Estonia made by Mark Kendall and published on beststartup.eu (Kendall, 2021). The fourth source was the list of Top Game Development Companies in Estonia on goodfirms.co (GoodFirms, 2022). The fifth source was a short list of collaborating companies that the Computer Graphics and Virtual Reality Study Lab at the University of Tartu had listed on their webpage (CGVR Lab, 2022). The final source was the member list of GameDev Estonia, which is a relatively new grouping organization representing many Estonian video game development companies (GameDev Estonia, 2022). These sources also included newly established companies, which were important to include as they represent the field from the perspective of startups.

The companies on the lists from these six sources were then further researched. We looked to see if the company had a valid entry in the credit registry E-krediidiinfo (http://e-krediidiinfo.ee) and if they had some internet presence to indicate that they are still an active company. In total, there were 70 companies deemed potentially active and thus valid participants of the survey.

3.3 The Survey

The chosen participants were sent a link to the survey to their official email found in the official business entry in E-krediidiinfo credit registry. If another email address was found on the company website or similar source, the given address was used in combination with the official one. The first email was sent on 4th of April, and if the company had not participated in the survey yet, the follow-up emails were sent on 18th and 26th. The emails explained that the survey is used to develop a Bachelor’s level video game designer-developer curriculum. The participants were offered a free overview of the survey results. They were also informed that answering the questionnaire indicates that they allow their answers to be used anonymously in research. The data were collected for one month.

4. Results

In total 28 different companies answered the survey. A few explicitly declined as they did not consider themselves as the correct target audience. A few also filled the survey multiple times, in which case we contacted them and agreed on which of the answers they deemed the most accurate.

4.1 Knowledge Areas

Compared to the results from the survey by McGill (2009), there were a few noticeable differences (see Figure 1). We performed the Welch’s t-test of unequal variances to determine if an item was rated statistically different ($\alpha=0.05$). The differences were found with Optimization (average 0.81 points higher in our survey, $p=0.0017$) and with Version Control Processes (average 0.73 points higher in our survey, $p=0.0140$).

Our added items Video Game Design, Video Game Analysis, and Software Testing were rated with averages of 4.36, 3.96, and 3.93 respectively. Ordering by the mean, these result in the 2nd, 9th, and 10th ranks.

4.2 Languages

In the *Languages* category we observed several differences (see Figure 2). There was an increase in the average importance of JavaScript (+0.80 points, $p=0.031$) and C# (+0.75 points, $p=0.031$), while a decrease of importance was observed with C++ (-1.07 points, $p<10^{-4}$), C (-0.98 points, $p=0.008$), and XML (-0.73 points, $p=0.043$). It is worth noting that while the largest decrease was with C++, it still ranked third in our results.

Our added items Visual Languages, JSON, GLSL / HLSL, TypeScript, Go, and Rust were rated with averages of 4.07, 3.75, 2.64, 2.50, 2.29, and 2.07 respectively. Ordering by the mean, these result in the 2nd, 4th, 13th, 15th, 17th, and 18th ranks.

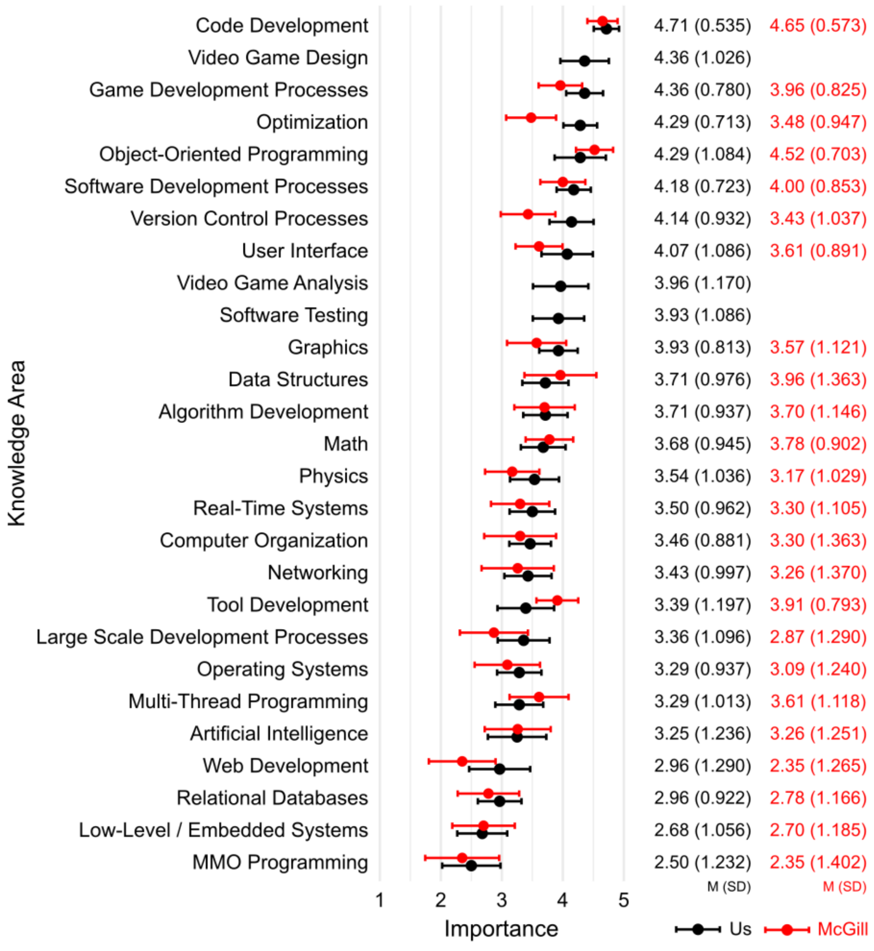

Figure 1. Our and McGill survey results for the *Knowledge Area* category.

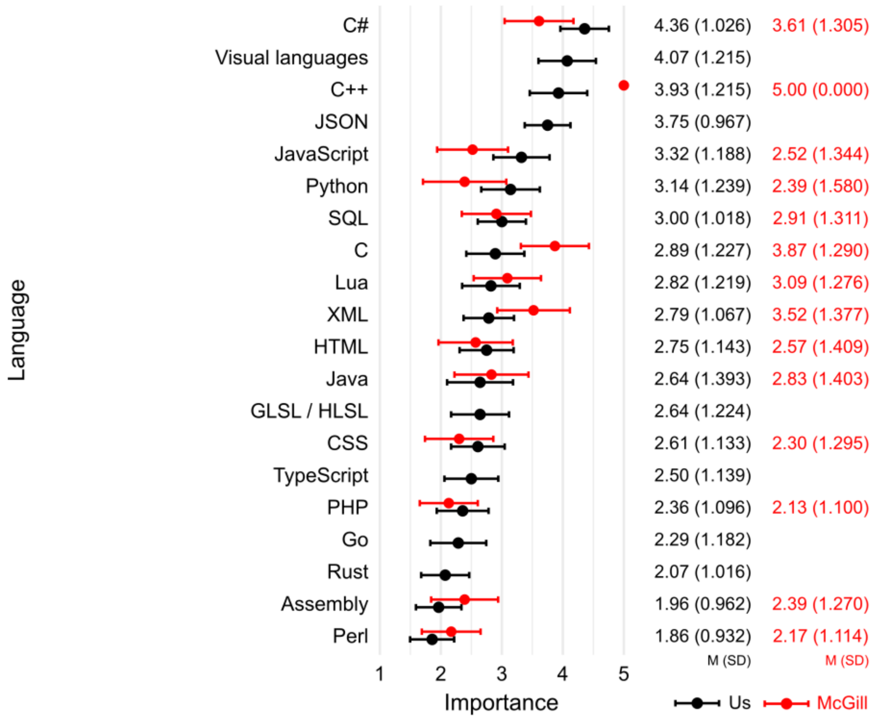

Figure 2. Our and McGill survey results for the *Language* category.

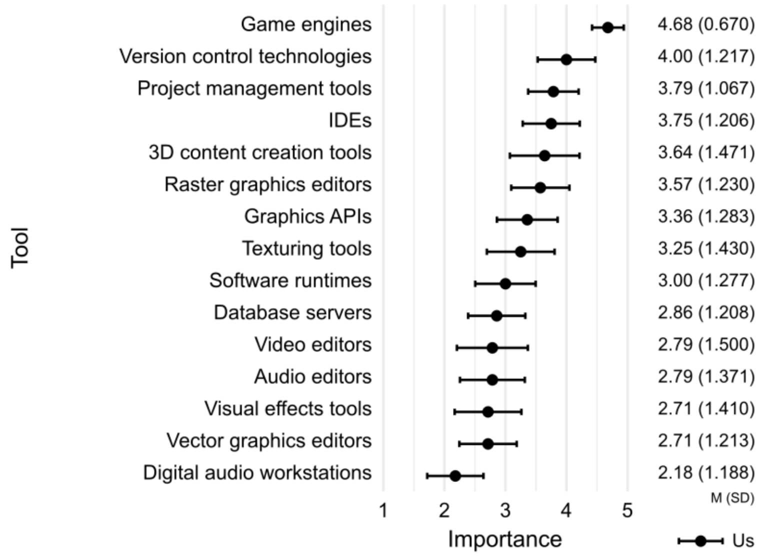

Figure 3. Our survey results for the *Software Tool / Environment* category.

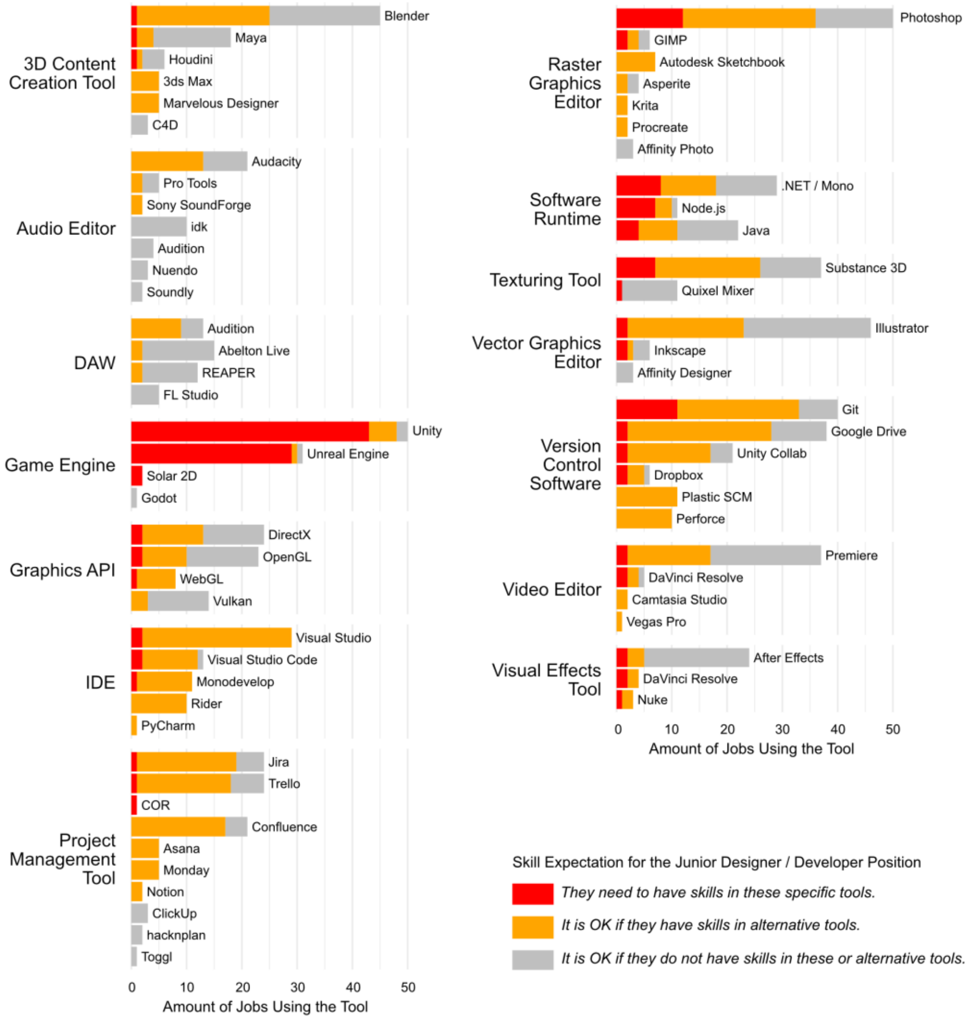

Figure 4. Usage and skill expectations for individual tools in the surveyed companies, scaled by the number of estimated annual employment positions for the junior designer / developer job.

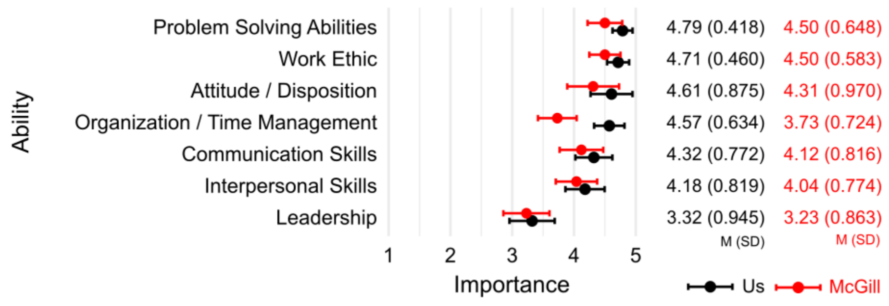

Figure 6. Our and McGill survey results for the Ability category.

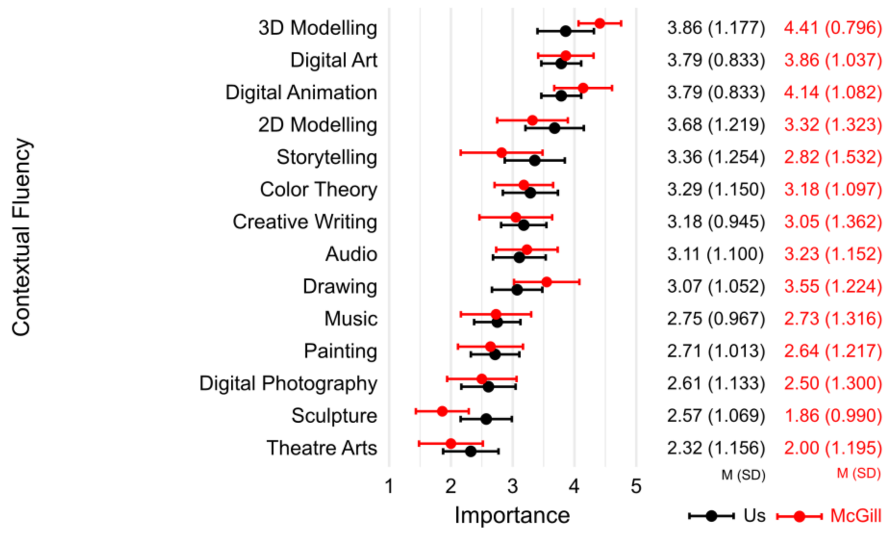

Figure 5. Our and McGill survey results for the Contextual Fluency category.

4.3 Software Tools / Environments

As the *Software Tools / Environments* category was completely reworked for our survey, a direct comparison with the McGill results cannot be made. It is still noteworthy that in our results the Game Engines item has the first rank with a mean of 4.68 and a relatively small standard deviation of 0.670 (see Figure 3). Perhaps with the exception of Director and Flash, the survey by McGill did not feature game engines at the time. This is understandable as, while there existed some game engines like GameMaker, a more substantial rise of game engines came in the mid-2010s (Andrade, 2015).

In our survey, all the 28 companies answered that they would every year hire in total 53 employees who would fit their hiring criteria for a video game designer / developer position. We used that number of annual new employees for each company to scale the answers to the questions that asked for the use and necessity of individual tools. This gives a clearer picture of the actual needs in the job market (see Figure 4).

We found that the most specific needs were in the game engines Unity (43/53) and Unreal Engine (29/53). These were followed by the raster graphics editor Photoshop (12/53) and version control software Git (11/53).

4.4 Abilities

The *Abilities* category (see Figure 5) had only one statistically significant difference. The Organization / Time Management item was rated higher (+0.84 points, $p<10^{-4}$).

4.5 Contextual Fluencies

There was also only one statistical difference in the *Contextual Fluencies* category (see Figure 6). This was the item Sculpture, which was rated higher in our survey (+0.71 points, p=0.017).

5. Discussion and Conclusion

The results of our survey are in large part similar to the results by McGill from 2009. This leads us to conclude that the needs of the beginning Estonian video game development industry are comparable to the needs of the well-established industries in the US and Canada. The implication of this is that graduates of a curriculum designed after these results could find employment with industries in different stages of growth.

The new findings are somewhat expected. In the *Knowledge Areas* category, the rise of importance in Optimization could be due to the rise in the market share of mobile games (Wallach, 2020). It could also be due to the fact that developers use game engines not being careful with performance management as the engine hides a lot of the underlying complexity from them.

The importance of the Version Control Processes was also significantly higher. This could be due to the rise of the popularity of version control technologies, namely Git, with the launch of GitHub in 2008 and its overwhelming success (Kashyap, 2020).

It is notable that our added items Video Game Design, Video Game Analysis, and Software Testing were rated quite important. We acknowledge that the original study focused on the video game developer position and our study extended this to a designer position. Items like Video Game Design might be important for a designer or a designer-developer, but not necessarily for a mere developer. However, as many curricula are a mix of designer and developer studies (Bouchrika, 2021; IGDA, 2008), then we feel this choice to be justified here. We think a video game designer should be able to write a quick prototype code and understand what is technically possible for the developer to implement. The developer should also understand game mechanics and user experience design and be able to clearly grasp the designer's intent. Thus, these two roles are highly coupled in practice.

The rise of C# and possibly the fall of C++ in the *Languages* category is most likely due to the great popularity of the Unity game engine, which uses C# as its language. The increase in the importance of JavaScript is possibly due to the advancement of web technologies. However, the importance of web development itself for a video game designer / developer was still rated quite low. Out of our added items, Visual Languages was rated quite important. This is likely due to the increase in the popularity of Unreal Engine, but also because of visual languages in other popular technologies, for example, Unity's Shader Graph, and content creation tools like Blender and Arcweave. On the other hand, the shader programming languages GLSL / HLSL were rated quite low in importance. This makes sense as game designers and regular game developers are not likely to program computer graphics in a shader. That would be done by game developers with an interest in computer graphics, or more accurately, computer graphics programmers.

From the answers to the questions about individual tools (Figure 4), it is evident that the game development companies need designers and developers who have specific knowledge about the Unity and Unreal Engine game engines. There were also many answers that listed both engines together, indicating that both are in use in their company, and skills in both are sought after.

There was a significant number of answers indicating the need for skills specifically in Adobe Photoshop. This implies that having skills in alternative (and cheaper, but similarly capable) tools like Affinity Photo would be insufficient for a significant proportion of the jobs.

In the *Ability* category, the item Organization / Time Management was rated more important than before. This could be due to a more widespread access to the internet, as more work is done remotely and thus employees are also expected to be more self-managed.

The results of our survey give a revised perspective on the needs of the video game industry compared to the previous study by McGill made over a decade ago. These results can help curriculum designers make more informed choices when establishing the learning outcomes and the use of specific tools in video game designer and developer curricula.

Acknowledgments

This research has been funded by the European Social Fund. We are grateful to all the video game development companies who participated in our survey and we hope our research helps you as well as curriculum designers.

References

Andrade, A. (2015). Game engines: A survey. *EAI Endorsed Transactions on Game-Based Learning*, *vol 2*, e8. https://doi.org/10.4108/eai.5-11-2015.150615

Anonymous. (2022). *Shipped.ee*. http://shipped.ee/

Bouchrika, I. (2021, June 25). *Video Game Design Degree Guide: 2022 Costs, Requirements & Job Opportunities*. Research.Com. https://research.com/degrees/video-game-design-degree

Computer Graphics and Virtual Reality Lab [CGVR Lab]. (2022). Companies. *The CGVR Lab*. https://cgvr.cs.ut.ee/wp/index.php/companies/

Estonian Virtual and Augmented Reality Association [EEVR]. (2022). *Estonian Virtual and Augmented Reality Association*. EEVR. https://eevr.ee/

GameDev Estonia. (2022). Database. *GameDev Estonia*. https://gamedevestonia.ee/location/

GoodFirms. (2022). *Top Game Development Companies in Estonia 2022 | GoodFirms*. https://www.goodfirms.co/game-development-companies/estonia

IGDA. (2008). *IGDA Curriculum Framework 2008*. https://docplayer.net/1868720-Igda-curriculum-framework.html

Kashyap, N. (2020, March 4). *GitHub's Path to 128M Public Repositories*. Medium. https://towardsdatascience.com/githubs-path-to-128m-public-repositories-f6f656ab56b1

Kendall, M. (2021, June 28). *33 Top Gaming Startups and Companies in Estonia (2021) – BestStartup.eu*. https://beststartup.eu/33-top-gaming-startups-and-companies-in-estonia-2021/

Kenwright, B. (2016). Holistic game development curriculum. *SIGGRAPH ASIA 2016 Symposium on Education*, 1–5. https://doi.org/10.1145/2993352.2993354

Mateas, M. (2007). *Design Issues for Undergraduate Game-Oriented Degrees*. https://www.semanticscholar.org/paper/Design-Issues-for-Undergraduate-Game-Oriented-Mateas/c4dd333871933a919cf2e3aac296e057d2b3941b

McGill, M. M. (2008). *Critical skills for game developers: An analysis of skills sought by industry*. 89–96. https://doi.org/10.1145/1496984.1497000

McGill, M. M. (2009). Defining the expectation gap: A comparison of industry needs and existing game development curriculum. *Proceedings of the 4th International Conference on Foundations of Digital Games*, 129–136. https://doi.org/10.1145/1536513.1536542

Mikami, K., Watanabe, T., Yamaji, K., Ozawa, K., Ito, A., Kawashima, M., Takeuchi, R., Kondo, K., & Kaneko, M. (2010). Construction trial of a practical education curriculum for game development by industry–university collaboration in Japan. *Computers & Graphics*, *34*(6), 791–799. https://doi.org/10.1016/j.cag.2010.09.015

Rogers, S. (2014). *Level Up! The Guide to Great Video Game Design* (2nd edition). Wiley.

Salen, K., & Zimmerman, E. (2003). *Rules of Play: Game Design Fundamentals*. MIT Press.

Schell, J. (2008). *The Art of Game Design: A book of lenses*. CRC Press.

Swink, S. (2008). *Game Feel: A Game Designer's Guide to Virtual Sensation* (1st edition). CRC Press.

Valentine, R. (2014, September 11). *Video game degrees in the US increase by 50% in five years*. GamesIndustry.Biz. https://www.gamesindustry.biz/articles/2014-09-11-video-game-degrees-in-the-us-have-increased-by-50-percent-over-the-last-five-years

Wallach. (2020, November 23). *50 Years of Gaming History, by Revenue Stream (1970-2020)*. https://www.visualcapitalist.com/50-years-gaming-history-revenue-stream/