Assignment 2

Dong min Kim
dkim614@gatech.edu

1 RESEARCH LOG

1.1 Background

Last week I studied many papers about intelligent tutoring systems, and I found 4 big ideas: learning step-by-step, mastery-based, adaptive feedback, and thinking about your own thinking (metacognition). Now I want to try make my SQL learning app more fun and more real-life feeling. This week, I looked at papers about problem-based learning (PBL) and game-based learning (GBL). I wanted to learn how to make learning more interesting, especially for beginners who think SQL is boring or hard. Maybe putting story or mission inside can help user focus and keep going.

So this week, I searched papers about scaffolding (step-by-step help), transformational play (user becomes main character in a story), and gamification (add game style like points, goals). I think these ideas can help my app not only teach but also keep students interested and not quit early.

1.2 Papers

1) Transformational Play as a Curricular Scaffold (Barab et al., 2009)
This paper talks about how students learn better when they are part of a story. The students play a role and solve problems in a game-like world. This idea gave me the thought that my SQL app can have a mission or story to keep users more focused and motivated. It made me realize learning feels stronger when it's inside a meaningful context.

2) Scaffolding and Achievement in Problem-Based and Inquiry Learning (Hmelo-Silver et al., 2007)
This paper says problem-based learning works better when students get help step by step. This help is called scaffolding. For my app, it means I should not

give answers too fast, but let the learner try first and give help after they make a mistake. That way, students can feel like they are learning, not just copying.

3) Scaffolding Teachers' Efforts to Implement Problem-Based Learning (Ertmer & Simons, 2006)
Even teachers need help to use PBL correctly. They need tools, example lessons, and support. For my app, maybe I can give users mini goals or small steps so they don't feel confused. This makes it easier for learners to build confidence as they go.

4) Not just fun, but serious strategies (Kim et al., 2009)
This paper says games are good for learning when they include meta-cognitive strategies. So students have to think about their thinking. I think I can ask users to write short reason why they wrote that SQL query before I give them feedback. That can help users understand their own logic and fix problems better.

5) Problem-Based Learning (Hung et al., 2008)
This paper explains what PBL is. Students learn better when solving real-world problems. It made me think my SQL app should have tasks with real data or real situations, not just boring examples. Solving real problems can also make learning feel useful and rewarding.

6) When is PBL More Effective? (Strobel & van Barneveld, 2009)
They looked at many PBL studies and found that PBL helps students learn better for the long term, not always for tests. This supports the idea that my app can help users understand deeply, not just memorize. PBL also increases motivation because students work on something that matters.

7) Entering the Education Arcade (Jenkins et al., 2003)
This paper talks about how games can be good for education. It gave examples where students were more engaged. That made me feel good about mixing SQL learning with light game features. It shows that learning can be fun and effective at the same time.

8) Questioning Video Games! (DiSalvo & Bruckman, 2011)
This paper shows how games can help more students, even students who don't like CS before. But the game must connect to real CS learning. I will make sure my app teaches real SQL skills, not just has fun parts. The game should help them see how CS is useful, not just entertaining.

9) Game-Based Learning in Science Education (Li & Tsai, 2013)
This paper reviews many studies and says game-based learning can improve motivation and learning if the game has clear goals and feedback. I want to make sure my app gives clear results after each SQL task. Students learn better when they know what they did right or wrong.

10) Environmental Detectives (Klopfer & Squire, 2008)
They made an AR game where students solve real-world problems. It helped students feel like scientists. I can't use AR, but I can still design tasks that feel like real jobs for data analysts. This approach can make SQL tasks feel exciting instead of dry.

11) Intelligent Tutoring and Games (McNamara et al., 2009)
They combined games and tutoring systems and showed that it helped students stay more focused and learn more. This fits my idea well — I want my app to be like a mix of tutor and game. The tutoring system keeps users on track while the game parts keep them engaged.

12) The Value of Serious Play (Rieber et al., 1998)
Play can be serious and help learning. People take more risks and try more when learning feels playful. I want my app to let people try things without being afraid of being wrong. This makes learning SQL less stressful and more enjoyable.

13) Distinguishing between games and simulations (Sauvé et al., 2007)
Games and simulations are not the same. Games have rules and goals, simulations show real things. Maybe I can mix both — show SQL as real work and also add goals. It helps learners practice in real-world style but also keep track of their progress.

14) Why Gamification is Bullshit (Bogost, 2011)
This paper says just adding points and badges is not real learning. The game part must be meaningful. So I will not add reward just for fun — it should help user learn SQL better. This reminds me to focus on learning value, not just shiny features.

15) Digital Game-Based Learning in High School CS Education (Papastergiou, 2009)
Games helped high school students learn computer science better. They liked it and did better than normal classes. This gave me confidence that my SQL app can really help students too. It also means game-based learning is not just for young kids — it works for CS learners too.

1.3 Synthesis

This week I saw many papers talk about how to help students learn better by giving support step-by-step, not just let them figure out everything. This idea is similar with what I read before in VanLehn (2006) about tutoring loops. But now I saw it used more in real school and real class, like science game or museum activity. I learn that when students feel they are doing something real (like solving problem as a scientist or analyst), they try more and understand deeper.

Another idea is game and story make students feel it's fun, not just homework. In Barab et al. (2009) they talk about “transformational play” where student becomes main person in the story. That gave me idea for SQL app to maybe tell story like "You are data scientist, help the team find the answer." I also liked how papers say we must be careful — not all game idea is good. Some like badges or points don't help if student don’t feel the learning is useful (Bogost, 2011).

I feel like now I have more idea how to combine these — game story can help keep motivation, scaffolding helps not get lost, and adaptive feedback makes it feel personal. All of these match with my app goal.

1.4 Reflection

At first, I only focused on how to make my SQL app work correctly — how to give the right answers and explain the logic clearly. But after reading the papers, I started thinking about how to make the learning experience actually enjoyable. I remembered how I used a coding app in the past but stopped because it felt boring. There was no feedback, no clear sense of progress, and no reason to keep going.

Now I understand that even small elements like missions, simple stories, or visible goals can help learners stay motivated. Some of the papers were hard to understand at first because they included a lot of academic terms and theories, but I took time to reread and translate them into my own words. That process helped me realize that not all games automatically lead to learning. A good learning game needs to be designed with purpose — not just with flashy features.

Through this assignment, I've learned that I don't need to create a huge or complicated game. What really matters is offering learners small missions, clear feedback, and a meaningful context. That alone can make a big difference in helping users stay engaged and feel like they're really improving.

1.5 Planning

Next week, I plan to explore more about computer science education. Since my app is designed to help people learn SQL, which is often part of beginner-level computer science, I want to read papers about how to teach CS effectively. I'm also curious to see whether high school and college students have different learning needs, so I'll look into that too.

In addition, I want to sketch a simple design for the app — how each lesson will flow and what the screens might look like. I'm thinking about creating two learning paths: one for complete beginners and another for learners who already know some basic SQL. If I have enough time, I'd like to test an early version with one or two friends and ask which parts feel confusing or boring.

Lastly, I want to start reading about how large language models (like ChatGPT) are being used in education. I'm not sure if I'll include LLMs in my app yet, but in the future, they could be helpful for giving personalized feedback or acting as a tutor during practice.

2 ACTIVITY

The Paradox of Progress: How Generative AI May Reinforce Educational Inequity

Generative AI (GenAI), particularly large language models (LLMs), is rapidly transforming the educational landscape. It promises scalable, personalized instruction and unprecedented access to high-quality content. While these advantages are often celebrated, it is important to consider the deeper, systemic implications of GenAI deployment—especially as it pertains to educational equity. This paper argues that, rather than serving as a bridge to close learning gaps, generative AI has the potential to reinforce and even exacerbate existing educational disparities.

Technological Barriers and the Digital Divide

At the core of GenAI lies an implicit assumption: that students have reliable access to modern computing infrastructure and high-speed internet. However, this assumption fails in many low-income, rural, and underserved communities globally. While some basic features of AI-powered tools may be accessible via smartphones, more advanced educational tasks—such as rendering complex mathematical explanations or supporting interactive simulations—require high-performance devices and stable connections. Kasneci et al. (2023) note that "the high computational cost and infrastructural requirements pose a barrier to equitable implementation, especially in low-resource educational settings." This digital divide can manifest in multiple ways: students in affluent districts receive high-quality AI tutoring, while their counterparts in disadvantaged areas are left with minimal or outdated tools.

Commercialization and Tiered Access

Another critical barrier to equity is the commercialization of GenAI technologies. Many of the most powerful LLMs are proprietary and operate under paywall-restricted APIs or premium subscription models. The logic is simple: those who can pay gain access to more accurate, responsive, and personalized AI tools. As Rahman and Watanobe (2023) argue, this creates a "stratification in educational AI platforms, where the most capable tools are monetized and thus inaccessible to lower-income learners."

This two-tiered access model means that GenAI may inadvertently replicate the very disparities it aims to resolve. Wealthier families and institutions can afford to integrate GenAI into their curricula, providing students with a competitive edge. Meanwhile, students from under-resourced backgrounds might only experience free or outdated models with limited functionality and accuracy.

Loss of Human and Cultural Diversity in Learning

While technological access and cost are critical concerns, there is also a less tangible but equally significant dimension to educational equity: cultural and human diversity. Traditional classroom settings provide more than just content delivery—they offer mentorship, emotional support, and exposure to diverse perspectives. Teachers, especially those from varied cultural and socioeconomic backgrounds, help students connect abstract knowledge to lived experience.

Generative AI, in contrast, is trained on large datasets that often reflect Western, English-speaking, and majority-culture biases. Ahmad et al. (2023) caution that "Generative AI systems lack the socio-emotional intelligence and cultural responsiveness that human teachers bring to the classroom." For students from marginalized communities, education is not merely about acquiring knowledge; it is about seeing themselves reflected in the process of learning. Homogenized AI outputs threaten this form of representation, especially when students rely exclusively on AI-generated instruction.

Moreover, lower-quality AI systems, which are often the only ones available to disadvantaged learners, may reinforce shallow or stereotypical representations, further widening the cultural gap. The more we rely on GenAI for full-scale instruction without human integration, the more we risk eroding the diversity and richness that human educators bring to the table.

Conclusion

While generative AI offers tremendous potential to personalize education and expand access, its current implementation risks deepening the very inequities it claims to mitigate. Issues of digital access, commercialization, and cultural homogenization must be addressed if GenAI is to fulfill its promise equitably. Policymakers, educators, and developers alike must work to ensure that these technologies are accessible, inclusive, and embedded within a broader, human-centered pedagogical framework. Only then can generative AI become a true force for educational equity.

3 REFERENCES

1. Barab, S., Scott, B., Siyahhan, S., Holdstone, R., Ingram-Goble, A., Zuiker, S., & Warren, S. (2009). *Transformational play as a curricular scaffold: Using videogames to support science education*. Journal of Science Education and Technology, 18(4), 305–320. https://doi.org/10.1007/s10956-009-9181-5
2. Bogost, I. (2011). *Why gamification is bullshit*. In S. Walz & S. Deterding (Eds.), The gameful world: Approaches, issues, applications (pp. 1–8). MIT Press.
3. DiSalvo, B., & Bruckman, A. (2011). *Questioning video games: Influence on computer science interest*. Proceedings of the 2011 ACM Annual Conference on Human Factors in Computing Systems (CHI '11), 1589–1598. https://doi.org/10.1145/1978942.1979175
4. Ertmer, P. A., & Simons, K. D. (2006). *Scaffolding teachers' efforts to implement problem-based learning: Lessons from a longitudinal study*. Journal of Educational Computing Research, 33(4), 403–421. https://doi.org/10.2190/ERV4-79L4-281Q-1R15
5. Hmelo-Silver, C. E., Duncan, R. G., & Chinn, C. A. (2007). *Scaffolding and achievement in problem-based and inquiry learning: A response to Kirschner,*

Sweller, and Clark (2006). Educational Psychologist, 42(2), 99–107. https://doi.org/10.1080/00461520701263368

6. Hung, W., Jonassen, D., & Liu, R. (2008). *Problem-based learning*. In J. M. Spector, M. D. Merrill, J. J. G. van Merriënboer, & M. P. Driscoll (Eds.), Handbook of research on educational communications and technology (3rd ed., pp. 485–506). Lawrence Erlbaum Associates.
7. Jenkins, H., Klopfer, E., Squire, K., & Tan, P. (2003). *Entering the education arcade*. ACM Computers in Entertainment, 1(1), 17–17. https://doi.org/10.1145/950566.950591
8. Kim, B., Park, H., & Baek, Y. (2009). *Not just fun, but serious strategies: Using meta-cognitive strategies in game-based learning*. Computers & Education, 52(4), 800–810. https://doi.org/10.1016/j.compedu.2008.12.004
9. Klopfer, E., & Squire, K. (2008). *Environmental detectives—the development of an augmented reality platform for environmental simulations*. Educational Technology Research and Development, 56(2), 203–228. https://doi.org/10.1007/s11423-007-9037-6
10. Li, M.-C., & Tsai, C.-C. (2013). *Game-based learning in science education: A review of relevant research*. Journal of Science Education and Technology, 22(6), 877–898. https://doi.org/10.1007/s10956-013-9436-x
11. McNamara, D. S., Jackson, G. T., & Graesser, A. C. (2009). *Intelligent tutoring and games (ITaG)*. In D. H. Jonassen & S. M. Land (Eds.), Theoretical foundations of learning environments (2nd ed., pp. 221–241). Routledge.
12. Papastergiou, M. (2009). *Digital game-based learning in high school computer science education: Impact on educational effectiveness and student motivation*. Computers & Education, 52(1), 1–12. https://doi.org/10.1016/j.compedu.2008.06.004
13. Rieber, L. P., Smith, L., & Noah, D. (1998). *The value of serious play*. Educational Technology, 38(6), 29–37.
14. Sauvé, L., Renaud, L., Kaufman, D., & Marquis, J. S. (2007). *Distinguishing between games and simulations: A systematic review*. Educational Technology & Society, 10(3), 247–256.
15. Strobel, J., & van Barneveld, A. (2009). *When is PBL more effective? A meta-synthesis of meta-analyses comparing PBL to conventional classrooms*.

Interdisciplinary Journal of Problem-Based Learning, 3(1), 44–58. https://doi.org/10.7771/1541-5015.1046
16. Ahmad, N., Murugesan, S., & Kshetri, N. (2023). Generative Artificial Intelligence and the Education Sector. *American University of Sharjah*.
17. Kasneci, E., Sessler, K., Bannert, M., Dementieva, D., Gasser, U., Groh, G., ... & Kuhn, J. (2023). ChatGPT for good? On opportunities and challenges of large language models for education. *Technical University of Munich & LMU Munich*.
18. Rahman, M., & Watanobe, Y. (2023). ChatGPT for Education and Research: Opportunities, Threats, and Strategies. *Dhaka University of Engineering & Technology*.