# 1 Research Context and Problem

Recently, educators and computer science researchers have begun partnering to develop new pedagogical tools based on wearable technology. Wearable Learning (WL) provides students with the tools to play physically active math games using mobile devices, as well as design their own games through the WL Platform. In the project described in this proposal, we will identify what characteristics can be extracted from student game designs by creating a software tool to process JSON files of student-created game designs, and then cross-validating our results with game-design artifacts and coding schemes created by WL researchers. The JSON files capture Finite State Machine (FSM) game models, the representation of games within the Wearable Learning Platform, which students use to design games. Our proposal describes the motivation for the project, the research questions we aim to address, and the concrete plan for carrying out the project.

## 1.1 Games For Learning

K-12 educators have sought many ways of introducing new concepts to the classroom. Many have looked beyond traditional learning and have considered introducing the use of educational games to support learning. There is research which suggests that games emphasizing academic knowledge and skills, such as reading or math, succeed in teaching their intended content [13]. With the prevalence of technology, many educators are beginning to build upon traditional learning by implementing online game-based learning in classrooms. Game-based learning creates an environment where players can learn to collaborate in groups and a means of enriching previously acquired knowledge [5].

Researchers have even looked beyond game-based learning and have explored shifting the game-based learning landscape from having students play educational games to designing their own. In mathematics, for example, game-design allows students to build on and challenge their existing knowledge, engage in relevant and meaningful contexts, and develop connections among their ideas and real world contexts [9]. Game design can be a powerful learning tool for both teachers and students, as it provides a situation that combines practice and theory, and provides opportunities for discussion, reflection, and collaboration within a meaningful context [9]. More specifically, computational thinking has been recognized as a fundamental skill that allows students to abstract and decompose complex problems in any discipline [6]. Game design that involves computational thinking is especially beneficial because it allows students to enact practices related to problem decomposition and simulation, making sense of modular components and how they interact with each other, troubleshooting and debugging, as well as sense making and employing conditional logic [10].

## 1.2 The Wearable Learning Platform

### **1.2.1 Overview of Wearable Learning**

Wearable Learning enables students and teachers to create, manage, and play physically active and social multiplayer games without the need of prior programming experience, and the curriculum promotes computational thinking through game creation and play [4]. The Wearable Learning Platform (WL) is a web-based infrastructure that uses game creation and embodied cognition to teach students math concepts and computational thinking skills and give them early exposure to computing concepts [2]. Wearable Learning uses FSMs as a way for students to program the game design process and flow required in their games. Finite State Machines (FSMs) are computational models that can be used to stimulate phenomena, sequential logic, and the behavior of computer programs. They include a finite number of states and directional transitions between states [4]. They can be used to model games created by students and can be used to represent their logic flow as well as computational thinking.

### **1.2.2 The Game Design Process**

Through the platform, students are able to access the platform and take on different roles as Game Editors, Game Players, or Game Managers. Game Editors program their games using the Wearable Learning’s drag-and-drop, FSM-based programming language. Game Editors configure the states to display media such as images and text and indicate possible inputs to specify transitions between states such as button presses or text entries. They are also given the opportunity to run and debug their games in the built-in debugger. As Game Players, students reinforce their subject knowledge in the curriculum. Students may even wear their mobile devices, which allows them to complete physical activities, manipulate their environment, and handle game materials. They can enter the game by using a PIN provided by their manager and select a role to join the game. As Game Managers, teachers and students can browse the platform to find any open-access games complete with instructions and materials that have been created by students, teachers, or researchers. They can select and start games from any device with WiFi access, generating a PIN that can be used by the entire classroom.

### **1.2.3 The Wearable Learning Curriculum**

Before introducing the Wearable Learning Platform to students, teachers play an existing technology-based game and then create their own games to engage with the technology, understand how the games are played and created, and deepen their own computational and mathematical skills to aid their students in the game design process. Once the teachers get accustomed to the platform, they create their own games on WL, which they can then deploy to their own classrooms. Students then play the game(s) that their teacher developed. After playing, they develop their own math games, which allows them to deepen their knowledge of the math subject, and learn computational concepts required to create and debug games.

The WL uses a visual programming language that involves Finite State Machines to represent states and transitions. The states are visually represented through boxes, while the transitions are represented as arrows that resemble a flow chart. All of these states and transitions are created through a drag-and-drop editor, which allows teachers and students to create games through the platform easily.

Before programming the games, students are asked to express their game design ideas as drawings and narrations on booklets. Students draw “state” boxes to represent the game screens on a mobile phone, and arrows or “transitions” to represent concrete inputs that allow the player to transition from one state to another.

### **1.2.4 Prior Work With Wearable Learning**

These artifacts have been used previously in Wearable Learning research to identify cross-cultural differences in student game design in order to identify better ways to support student learning. During the study, researchers looked for four aspects within the game design: social and physical interactions (collaboration, competition, and physicality option), technological descriptors (dependence and incorporation), game characteristics (Game Descriptors, Progressive Levels, and End Goal), and game representation components (FSMD Rules, Physical Objects, Physical Space, and Timing) [8].

Findings from this study showed that U.S. students exhibited more individualist characteristics with more technology incorporation, competition, specific end goals, and timing, while Argentine students exhibited more collectivist traits by incorporating collaboration, having a more consistent expression of game end goals than the U.S. students, and having less timing components.

This work provided insight into how students’ culture can potentially influence how they design their artifacts in terms of game design through the Wearable Learning Curriculum, and can be used for further research on how the learning curriculum can support or hinder students from expressing their respective cultural values and thinking in the games they design.

## 1.3 Identifying Characteristics in Game Design

Every game developed by the students and teachers is stored in a JSON representation within the Wearable Learning Platform. Our goal is to analyze the student design artifacts and Finite State Machine data in order to determine what characteristics can be extracted from the JSON data, as well as identify any common game characteristics prevalent in student game designs. By analyzing student game designs from India, Argentina, and the United States, we can gain a deeper understanding of the factors that aid student game design, and better support student learning.

Currently, the process for extracting characteristics in student game design is done manually by analyzing the student design artifacts, which include Finite State Machine Diagrams, with “state” boxes that correspond to game screens on the phone and arrows that specify concrete inputs that game players have to enter to move from one game state to the next [12].

Researchers in Wearable Learning have developed a coding guide and a coding scheme to analyze the artifacts. The goal of the coding guide is to assess the types of benefits that the WL game design process and game play might have on computational thinking [12]. The purpose of the coding scheme is to use the qualitative analysis of the designed games and finite state machines to further understand students’ processes of computational thinking development as they work with embodied technologies that complement physically active games [12].

The coding scheme has been developed with the following domains: Game Characteristics, Technological Descriptors, Team Descriptors, Collaboration and Competition, Kinesthetics and Physicality, Mathematical Relevance and Game Representation Components, with each domain containing a subdomain for further assessment [12]. Rather than manually extracting information from written booklets, our research aims to find a way to automate this process by creating an algorithm through the Finite State Machine Library. We hope that our research will lead to less human error in the extraction process as well as potentially find certain characteristics that may not have been discovered in previous Wearable Learning research.

## 1.4 Research Problem

Given the growing user base of WearableLearning, the amount of data in the future can become too excessive. In order to solve this problem, we plan to investigate the following questions:

1. How can we automate the data analysis process?
2. What kinds of game-design characteristics can we extract from the FSM database that go in accordance with the coding manual?
3. How can we extract qualitative information from the FSM database?

In the sections below, we will discuss how we aim to approach these questions as well as how we will evaluate our plan for our proposed project.

# 2. Proposed Solution

Our goal is to automate the data analysis process by developing a machine learning algorithm to efficiently analyze the JSON dataset. Our data contains the student-created game designs as well as the final Finite State Machine representations of the games. The final Finite State Machines are compiled into data stored as JSON files. Some of the potential features that can be detected from the data are the loops, the conditionals, the team-based features (whether the games are team-based or not), and the cultures (cultural backgrounds like US, Argentina, or India). On the same note, the game designs artifacts data (i.e. booklets) also contains FSM and game descriptions, and we will analyze this data manually to use it as an evaluation tool.

## 2.1. Analyzing JSON Data

### 2.1.1. Analyzing the Computational Concepts

The JSON data contains various attributes which includes game names, team number, player number, state, transition and connection count, the ID of each state, the input ID and output ID (which are the states that the current state is connected to) for each state, and the game description. There are also attributes that correspond to computational concepts such as loops or conditional statements. They can be found and counted by checking the Input and Output ID of each state in the Finite State Machine. We plan to have a separate approach for attributes like the computational concepts (i.e finding and counting). The description of the game in each state, however, is stored as text and would require a more complex program to analyze. Therefore, we are planning to use natural language processing to analyze the text.

### 2.1.2. Analyzing the Game Descriptions

Our idea to analyze the game descriptions would be extracting the keywords using BERT [11], a machine learning framework for natural language processing. From then on, we can further analyze the keywords using a multilabel text classification model, meaning that we will distribute each game into specific classes. One of our aiding tools would be the Coding Guide provided by the Advanced Learning Technology Lab of the University of Massachusetts Amherst. The Coding Guide contains a list of seven main domains that should be examined to determine the patterns in games. The domains are Game Characteristics (the general features of the game), Technological Descriptors (how is technology used in the game), Team Descriptors (the team of the game), Collaboration and Competition (the level of collaboration and competition of the game), Kinesthetics and Physicality (the physical nature and intensity of the game), Mathematical Relevance (the relationship of Math and gameplay) and Game Representation Components (which medium is the game created on). Each domain contains various sub-domains that can be potential features for our program to examine.

From the suggested features in the Coding Guide, we can use them to create the classes for the analyzed texts. To better illustrate our explanation, we predict that games with keywords like “shapes”, “square”, or “circle” can have a Geometry characteristic, or games with many fraction numbers can have a “Number and Operations with Fractions” characteristic.

After each game is processed and classified, we can find the average number of occurrences for every characteristic as well as find the common characteristics in games created by students.

# 3. Evaluation Plan

Based on our proposed solution, we have come up with two main ways to extract the most common features and the average number of concepts and patterns by analyzing data in the JSON files and the hand-drawings. We collect the most common features by first analyzing the model we created by examining JSON data and later cross-checking with the game design artifact data that was manually analyzed by other researchers. By doing this, we automate the data analysis process using machine learning particularly when the amount of data becomes excessive. After that, based on the model that we created, we could determine characteristics that are prevalent in student game design.

## 

## 3.1 Manual Analysis

Manually, we will extract keywords from the game design artifact data and put them into specific categories such as Geometry characteristic, Math Operation characteristic, etc. The more specific and representative to the category the keywords, the better datasets we will create. Those keywords which are categorized will act as ground truth dataset for the machine learning model.

## 3.2 The Machine Learning Model

We will create a machine-learning model which will help us identify the loops, condition, geometry characteristics, math operation characteristics, etc. After pretreating the JSON file, with the aid of BERT[11], a framework for natural language processing, we will be able to extract keywords then categorize them into specific categories based on the Coding Guide provided by the Advanced Learning Technology Lab of the University of Massachusetts Amherst. After getting the dataset, we split them into 3 sets: training set (60%), testing set (20%), and validating set (20%). While training the model, we would like to avoid the problem of overfitting and underfitting - addressing the problem where the model performs well on the training set but poor generalization to other data and poor performance on the training set and poor generalization to other data respectively. Therefore, we will create a graph of error versus model complexity to get insight on whether the model created is good for generalization or prone to underfit or overfit. After getting a desired model, the model will be evaluated by looking into the accuracy, precision, and recall that the training process gives. After that, we will then cross-checking this result with the ground truth that we gained from manually analyzing from the game design artifacts. We will create several models and will compare them to each other by the percentage gained from cross checking it with the manually analyzed data - this will give us the classification model that gives the highest accuracy based on the validation set that we manually analyzed.

## 3.3 Further Analysis

If the model were successfully built, we will be able to find the shared aspects in the game design and computational thinking of the students such as Progressive Difficulty, Game Facilitation, Mathematical Utilization, Pictorial Representation, etc. Finding such aspects could help us find the features that are shared among the game design of K-12 students.

# 4. Timeline

* **Over the winter break:** getting to know data samples, learning BERT and different machine learning models and how to preprocess text, Javascript, and reading files/JSON data.
* Week 1 - 2 (2/6/23 - 2/17/23):
  + Divided into two groups
    - Group 1 will plan the methods for extracting the data, divide the members to do the tasks, and begin to manually extract the data for the ground truth
    - Group 2 will start analyzing the JSON data and start finding the justifiable machine learning algorithms and libraries to work with
* Week 3 - 4 (2/20/23 - 3/3/23):
  + Group 1: Reporting the progress of extracting game design artifacts - expected to extract one third over the total of game design artifact data
  + Group 2: Expected to finish extracting the JSON data and start working on the machine learning model
* Week 5 (3/6/23 - 3/10/23):
  + Group 1: Reporting the progress of extracting JSON data - expected to extract one half over the total of game design artifact data
  + Group 2: Working on building the machine learning model
* Week 6 (3/13/23 - 3/17/23): Spring Break
* Week 7 - 8 (3/20/23 - 3/31/23):
  + Group 1: Reporting the progress of extracting JSON data - expected to finalize the data to feed into the machine learning model
  + Group 2: Get to the debugging phase, making justifiable changes to the model and begin training the model by the end of the week
* Week 9 - 10 (4/3/23 - 4/14/23): Train the model and write a report on the model’s performance
* Week 11 - 12 (4/17/23 - 4/28/23): Train the model and write a report on the model performance
* Week 13 - 14 (5/1/23 - 5/12/23): Find the shared aspects in the game design
* Week 15 - 16 (5/15/23 - 5/25/23): Finalize the report to submit
* **Submit by May 25th**

# 5. References

[1] Castro, Francisco. “WearableLearning Game Creation and Play Curriculum.” OSF, 11 July 2022. Web.

[2] Ivon Arroyo, Matthew Micciollo, Jonathan Casano, Erin Ottmar, Taylyn Hulse, and Ma. Mercedes Rodrigo. 2017. Wearable Learning: Multiplayer Embodied Games for Math. In Proceedings of the Annual Symposium on Computer-Human Interaction in Play (CHI PLAY '17). Association for Computing Machinery, New York, NY, USA, 205–216.

[3] Francisco Enrique Vicente Castro. 2022. Exploring the Use of Finite-State Machines and Game Creation to Teach Computational Thinking in Middle Schools. In Proceedings of the 27th ACM Conference on Innovation and Technology in Computer Science Education Vol. 2 (ITiCSE '22). Association for Computing Machinery, New York, NY, USA, 618.

[4] Arroyo, I., Closser, A.H., Castro, F. *et al.* The WearableLearning Platform: A Computational Thinking Tool Supporting Game Design and Active Play. *Tech Know Learn* (2022).

[5] Domna Chiotaki and Kostas Karpouzis. 2020. *Open and Cultural Data Games for Learning.* In Proceedings of the 15th International Conference on the Foundations of Digital Games (FDG '20). Association for Computing Machinery, New York, NY, USA, Article 107, 1–7.

[6] Jeannette Wing. 2009. Computational thinking. J. Comput. Sci. Coll. 24, 6 (June 2009), 6–7.

[7] Ivon Arroyo, Sai Gattupalli, Francisco Enrique Vicente Castro, Olivia Bogs, and Grace Seiche. 2022. “Exploring Cross-cultural Differences in Student Game Designs for WearableLearning”. International Society of the Learning Sciences. DOI: https://osf.io/jdsnt

[8]Finite State Machines. *Brilliant.org*. https://brilliant.org/wiki/finite-state-machines/

[9] Kafai, Y., Franke, M., Ching, C. *et al.* Game Design as an Interactive Learning Environment for Fostering Students' and Teachers' Mathematical Inquiry. *International Journal of Computers for Mathematical Learning* 3, 149–184 (1998). https://doi.org/10.1023/A:1009777905226

[10] *Aditi Wagh, Brian Gravel, and Eli Tucker-Raymond. 2017. The Role of Computational Thinking Practices in Making: How Beginning Youth Makers Encounter & Appropriate CT Practices in Making. In Proceedings of the 7th Annual Conference on Creativity and Fabrication in Education (FabLearn '17). Association for Computing Machinery, New York, NY, USA, Article 9, 1–8. https://doi.org/10.1145/3141798.3141808*

[11] Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.

[12] Castro, Francisco. 2022. “WearableLearning and Culture.” OSF. November 4. osf.io/yq49e.

[13] Blumberg, F.C., Deater-Deckard, K., Calvert, S.L., Flynn, R.M., Green, C.S., Arnold, D. and Brooks, P.J. (2019), Digital Games as a Context for Children's Cognitive Development: Research Recommendations and Policy Considerations. Soc Policy Rep, 32: 1-33. <https://doi.org/10.1002/sop2.3>

1. **Research Context and problem statement:** (Julia)
2. **Proposed Solution**: What is your specific proposed solution to the problem you identified in part 1? (Anh Pham)

2. **Evaluation Plan**: How will you know if your research was successful? (Minh)

1. **Timeline**: How will you organize your work to get it done in Spring? (Minh)
2. **References:** List of your references in ACM format. (Julia)