Workshop 1

Henry Ricaurte Mora 20221020084
Germán Darío Aya Fuentes 20232020091
Javier Alejandro Penagos Hernández 20221020028

Context

Game-based learning offers a fun and dynamic way of teaching, allowing students to interact with educational content in playful environments. Although its use has grown, there are still few open datasets that enable the application of data science to optimize these platforms. In particular, knowledge tracing, which allows for personalized learning, has been little explored in educational games.

The Kaggle competition is organized by Field Day Lab, a public lab from the Wisconsin Center for Education Research, which develops free and accessible games to study how people learn. In collaboration with organizations like The Learning Agency Lab, they aim to improve educational games and provide useful analytical tools for educators. If successful, these initiatives could expand and strengthen the use of game- based learning in formal education.

Components:

- **Mouse**: It is the first input. It is necessary for the functioning of the analyzed system.

- **Hover**, **Click**: Functions that are part of the first input.

- **X**, **Y**: Cartesian location of each of the inputs within the system.

- **Event_handler**: Group responsible for handling each of the individual events of the system.

- **Name**: Name of each event in the system.

- **Text**: Text associated with each event in the system.

- **Session_id**: Unique identifier of the session.

- **User_id**: Unique identifier of the user within the system.

- **Level**: User parameter that characterizes their skill level and/or progress.

- **Login_interface**: Interface that functions as an input to access the system.

- **Auth_system**: Authentication system to determine whether access is granted or denied.

- **Configuration**: Component responsible for managing the game's configurations.

- **Hq**: Identifier related to the visual configuration of the system.

- **Full_screen**: Identifier related to the visual configuration of the system.

- **Option_music**: Component that balances the game's music.

- **Screen**: Specifies characteristics related to the program's magnitude.

- **Audio_system**: External system component that provides the user with specific game audio.

- **Music_room**: Component responsible for the music in the room.

- **Music_game**: Component responsible for the game's music.

- **Id_notebook**: Unique identifier of the notebook.

- **Pages**: Space for making notes, paraphrasing ideas, or building a solution.

- **Event_registry**: Registry of all events in the system.

- **Data_analystic**: Component responsible for generating and managing the system's solution to reach the objective.

- **Performance**: Main output of the system (not feedback-based).

Relationships

Our first relationship is the one related to the **login_interface**, which leads us to an **auth_system** that is responsible for either returning a credential error or performing a login and modeling the system based on a **user_id**.

As a primary relationship, we have the one that generates the input: the **mouse**, which contains two main events: **hover** and **click**, where each of these is mapped using a Cartesian coordinate (**x**, **y**). This first "major" system constantly communicates by sending all information to the **event_handler**, which contains a **name** and a **text** for each of these events.

Continuing with the design of the user interface system, after entering the **user_id**, it obtains a **level**, which is responsible for setting in relation to the **user_id**. This same **user_id** component implements a **configuration** and creates a **session_id**, which is then responsible for sharing that information with the **event_handler**.
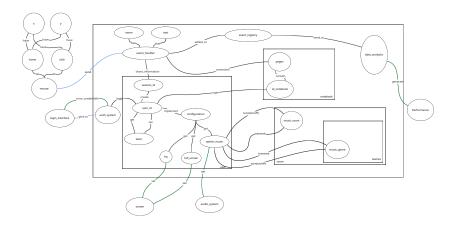
Approaching the **configuration** section, we have 3 relationships that obtain state information and share it with the corresponding components: **hq**, **full_screen**, and **option_music**. Both **hq** and **full_screen** send information to the **screen** as output.

Option_music also generates an output to the **audio_system**, and in addition, is able to receive what other components such as **music_room** and **music_game** transmit, and make changes by generating a turn **off** or **on**.

In **user_id** we also handle another relationship: this one gets to **id_notebook**'s component, which contains **pages** that are connected with an event we will manage as **event_handler**.

Event_handler is an important component since it adds everything it receives to another component called **event_registry**, which is in charge of handling all event records. These are then **sent_to** the **data_analystic**'s component, which generates a **performance** of the analyzed system.

Model

Complexity and Sensitivity

Sensitivity analysis allows us to understand how changes in one component of the system can affect the entire system. To do this, we analyze how each component influences the system and to what extent, with a particular focus on those components where sensitivity is highest—i.e., where variations in the output can most significantly impact the overall system. All of this is aimed at predicting player performance based on how these components behave.

First, we begin with the inputs. In the system, two inputs were defined.

Login Interface – Auth Login

The login interface is the user's entry point to access the game, and the auth login is the component that determines whether the user is allowed in or not. Although they do not directly affect the system, these elements are what enable the user to interact with the game itself.

Mouse Events

Events refer to the actions performed by the player using the mouse. These actions have a significant impact on the system, as depending on the decisions the player makes, how they make them, and the time taken to execute them, the system will respond in different ways. While there is not complete randomness—since in games actions are predefined—the variation lies in how the player carries them out.

Now that we have analyzed the inputs, we will analyze the system components that influence the system.

Level

The level reached by the player is a key element, as it reflects the progress and skills acquired during the game. However, it may present some limitations depending on the context, since this factor also largely depends on the player's ability, whether or not they use cheats to advance, and so on.

Session Id

The duration of each session per player allows for the analysis of whether players quickly become engaged with the game or, on the contrary, abandon it quickly. This factor is important for predicting player performance.

Configuration

Configuration is a very important aspect, as it can significantly influence the player's performance. Depending on the settings the player has adjusted, their in-game performance may vary in different ways. For the system presented, these settings include music volume, whether the game is in fullscreen mode, and the game's quality.

Data Analytics

This system component will analyze the outputs of the previously described components to identify patterns that help determine player performance. The limitations of this component are directly related to the inputs—therefore, if these are biased, the prediction will be as well.

Chaos and Randomness

A Complex and Dynamic System

Although the system appears deterministic and computational, in practice it behaves as a complex adaptive system. The diversity of individual trajectories, the influence of context, and the continuous feedback between the student and the environment make it a system whose evolution cannot be reduced to simple or linear rules.

These systems are characterized by the interaction of multiple variables, the emergence of behaviors, and sensitivity to small perturbations in the initial state. In the case of learning through games, the decisions, emotions, and particular conditions of each player interact with the system rules in unpredictable ways, generating highly variable dynamics that are difficult to model.

Chaotic Dynamics in Learning

A deeper analysis reveals characteristics of a chaotic system, understood as one that, although deterministic, exhibits unpredictable behaviors due to its extreme sensitivity to initial conditions. This is accompanied by high nonlinearity, the emergence of patterns, and the presence of strange attractors—all key concepts from chaos theory.

Sensitivity to Initial Conditions

Each student starts a game session with a set of personal and contextual variables: prior knowledge, motivation, emotional state, past experiences. Small variations in these factors can lead to completely different interaction trajectories with the system, making behavior very difficult to predict with certainty. This phenomenon is reminiscent of the famous "butterfly effect" in chaos theory.

Nonlinearity

The factors that influence performance do not have a direct or proportional relationship with results. For example, more time spent on a task may mean reflection in one case and confusion in another. This nonlinearity makes the system highly unpredictable and challenging to model.

Emergence and Attractors

As students interact with the system, emergent behavior patterns appear. Some learn through trial and error, others become frustrated and quit, and some internalize the rules and optimize their performance. These patterns can be understood as strange attractors—conceptual figures from chaos theory representing sets of states towards which a system tends without ever exactly repeating itself.

Chaotic Dynamics by Component

1. **Mouse Events (clicks, hover, x/y coordinates)**

   - *Chaotic nature:* Reflect micro-decisions by the student, influenced by attention, environmental interpretation, internal strategies, and emotional state.
   - *Randomness:* The same stimulus can provoke different responses depending on the individual's context.
   - *Impact:* Introduces noise into the system, complicating the direct association between event and performance.

2. **Response Time**

   - *Chaotic nature:* Affected by reading speed, fatigue, motivation, distractions, etc.
   - *Randomness:* A fast response may indicate confidence or guessing; a slow one may indicate reflection or confusion.
   - *Impact:* Needs contextualization for correct interpretation; its relation to performance is nonlinear.

3. **Level (player level)**

   - *Chaotic nature:* Depends on the accumulation of prior decisions and the player's adaptation to the environment.
   - *Randomness:* Two players at the same level may have followed completely different paths.
   - *Impact:* The level indicates relative progress, but not necessarily deep understanding.

4. **Session_id (session duration and frequency)**

- *Chaotic nature:* Influenced by personal and external factors, such as time availability or level of interest.
- *Randomness:* Sessions of the same duration may be qualitatively very different.
- *Impact:* Affects the amount of data per player and how their level of engagement is interpreted.

5. **Configuration (visual and sound options)**

- *Chaotic nature:* Interface variations change the sensory experience of the game.
- *Randomness:* Nonlinear impact on perception, concentration, or stress.
- *Impact:* Introduces heterogeneity into the environment that modifies player behavior.

6. **Event_handler and Event_registry**

- *Chaotic nature:* Represent unique and unrepeatable interaction sequences.
- *Randomness:* Each user follows a distinct path, with no exact replication.
- *Impact:* Key input for modeling, but require dimensionality reduction or grouping due to their complexity.

7. **Data_analytic and Performance**

- *Chaotic nature:* Dependent on the quality and representativeness of the inputs.
- *Randomness:* The model may detect misleading patterns or apparent relationships that do not reflect real connections if the data contains bias or noise.
- *Impact:* Limits the model's accuracy in trying to generalize in an inherently unpredictable system.

Conclusion

The system, based on the analysis performed, demonstrates a certain robustness in its relationships. This property ensures that its results are more precise and less susceptible to errors.

Although some uncertainty remains, it revolves around understanding the type of games it offers does the catalog exhibit a high and significant level of skill? These questions are necessary to provide a clearer understanding of the variability in configurations

We observe that the impact of each element is balanced in relation to both the system and its objective. While one component stands out such as **event_handler** and **event_registry** we can still note an overall balance in the importance weights.

Another important point is its internal configuration. Here, we can identify small subregions that improve diagram readability regions with significantly strong influence that, in addition to providing balance (as previously mentioned), offer control over the problem's surface and its purpose.