Happy Elements

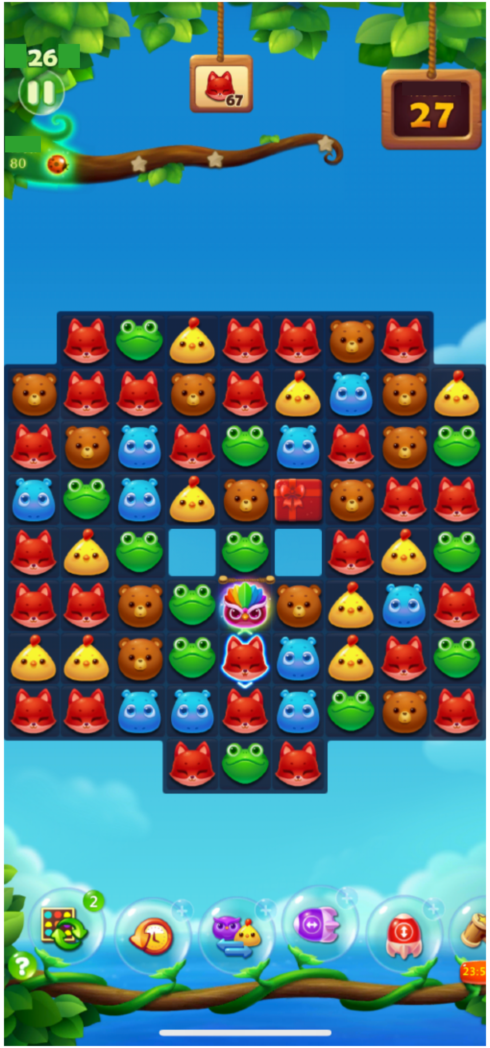
MAKE THE WORLD HAPPY

Generating the Best Game Experience through AI

Rein Houthooft

Happy Elements

Introduction

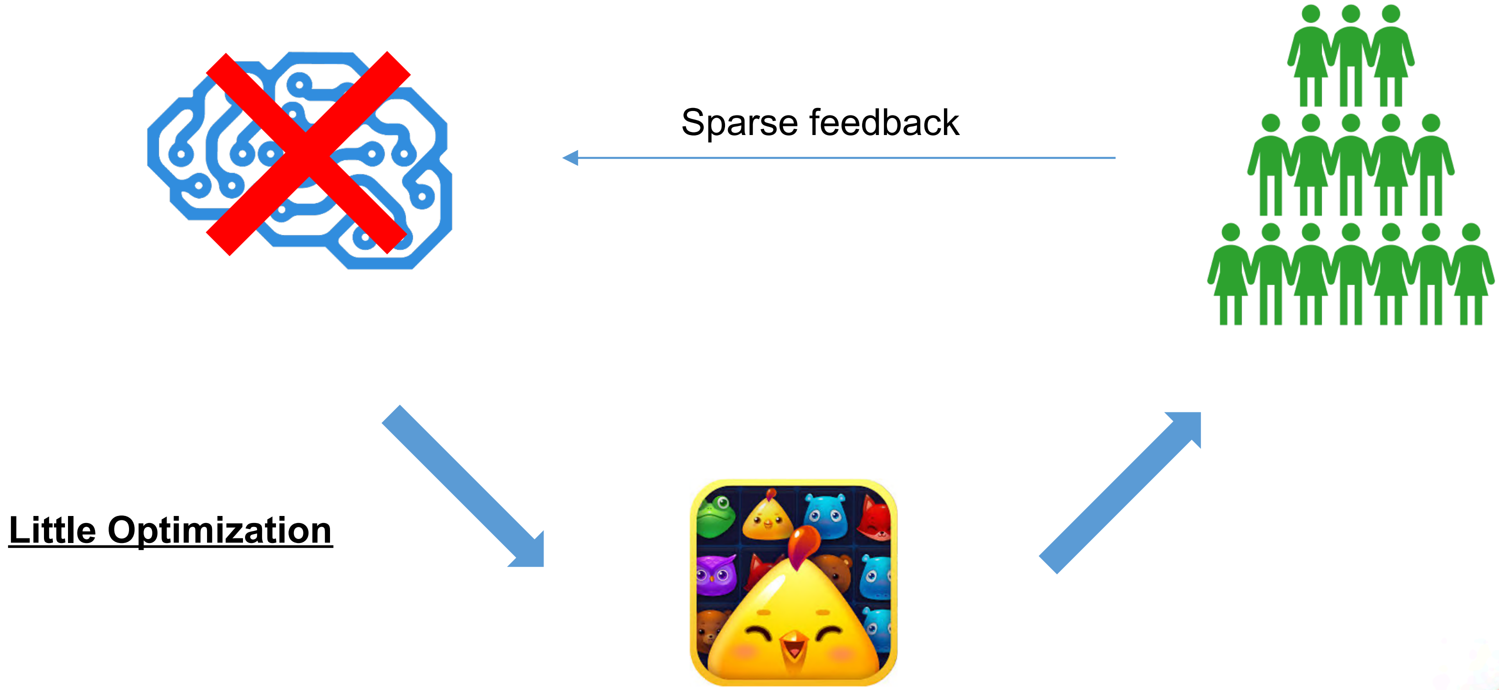


- Company focus: **casual** mobile games
- Main product: “**Anipop**” 
- Extremely popular (>**100M** users/month)
- Generates **TBs of data** each day
- **AI Lab** founded in 2018

Happy Elements

AI Lab: Goals & Strategy

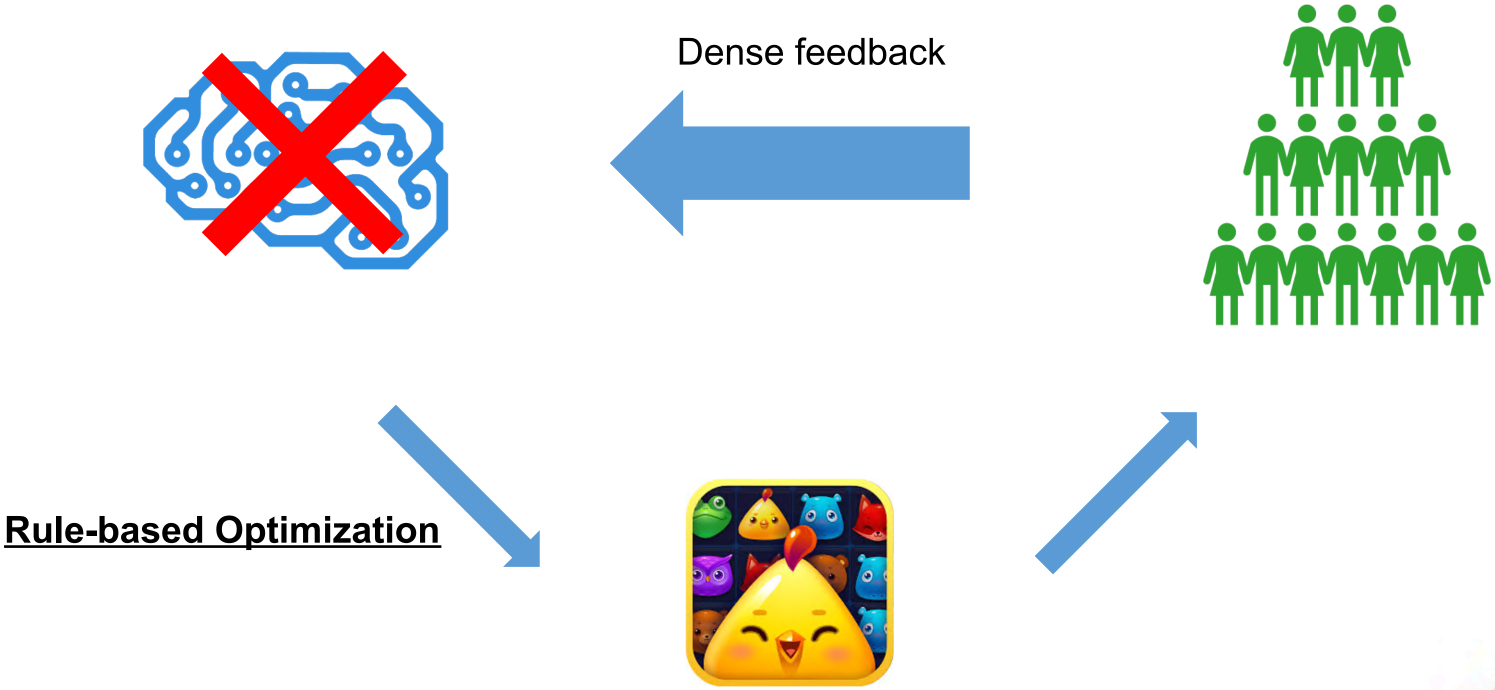
Open-loop to Closed-loop Game Design



Open-loop to Closed-loop Game Design

- Product-player preferences mismatch
- Designers slow in adapting to changing player behavior
- Lower player satisfaction

Open-loop to Closed-loop Game Design



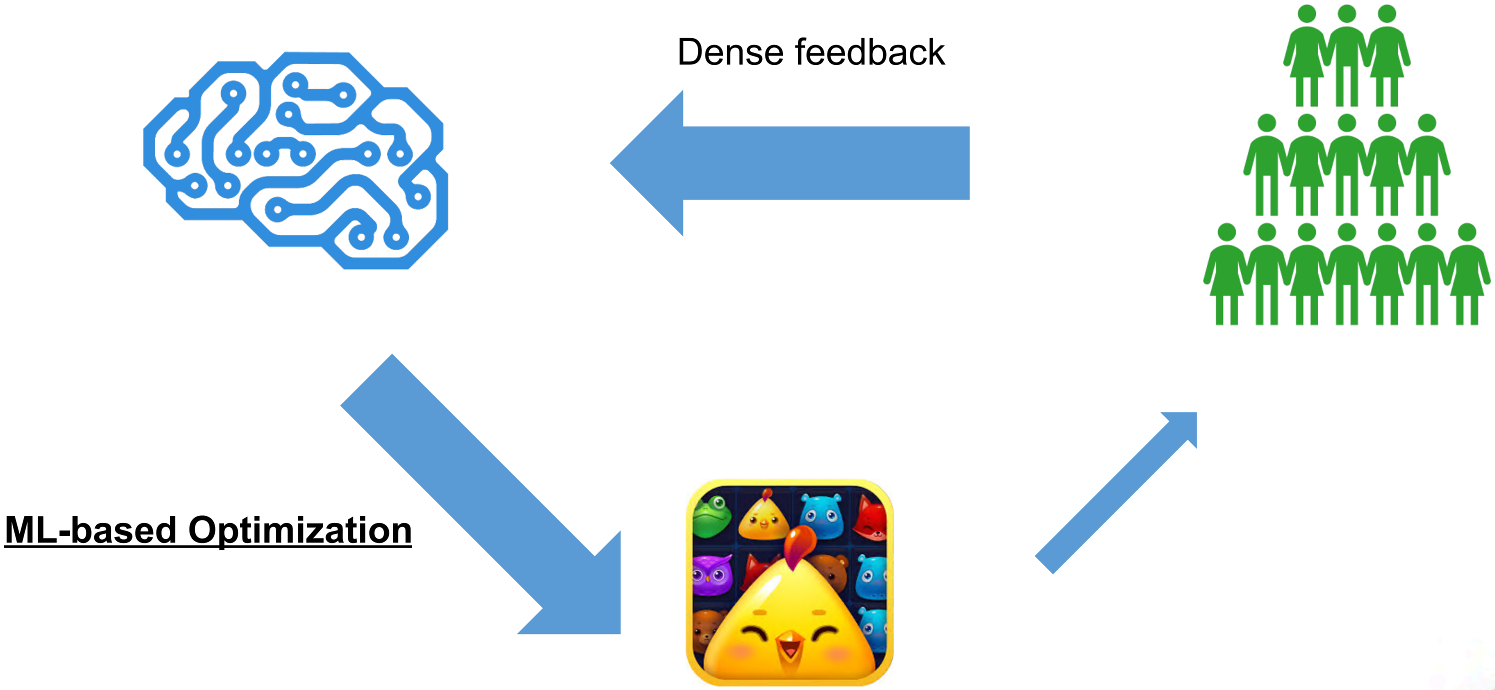
Happy Elements

Open-loop to Closed-loop Game Design

- Some adaptation to changing player preferences
- Low granularity
- Hard to maintain over time

Happy Elements

Open-loop to Closed-loop Game Design



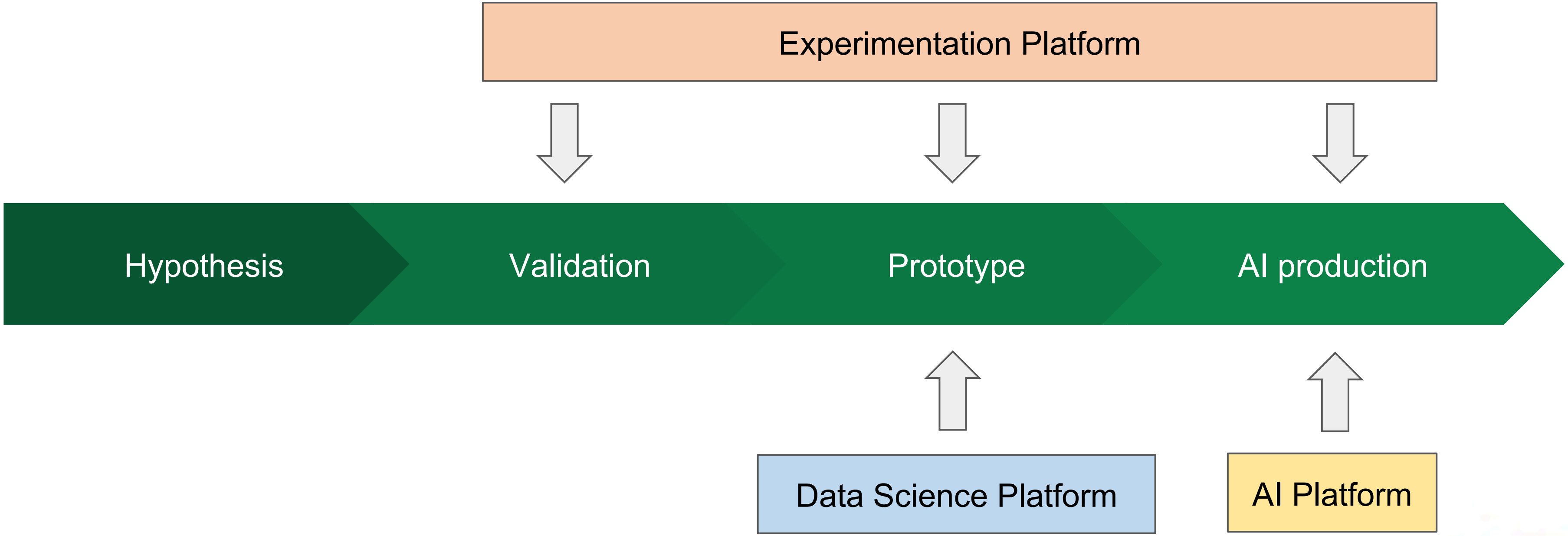
Open-loop to Closed-loop Game Design

- Immediate adaptation to changing player preferences
- High granularity
- Maintains itself through objective function optimization

One of our projects:

Deep Learning for Game Difficulty Adjustment

From Hypothesis to Production



Happy Elements

Game Difficulty Adjustment

- Hypothesis: **preferred** difficulty **varies** across users & time.
- Validation: difficulty correlates with **LTV/retention**.
- Prototype + Production: **adjust** difficulty dynamically via **ML**.

Problem Formulation

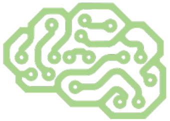
Happy Elements

Problem Formulation

Fixed gameplay
for **all** users

manual rules

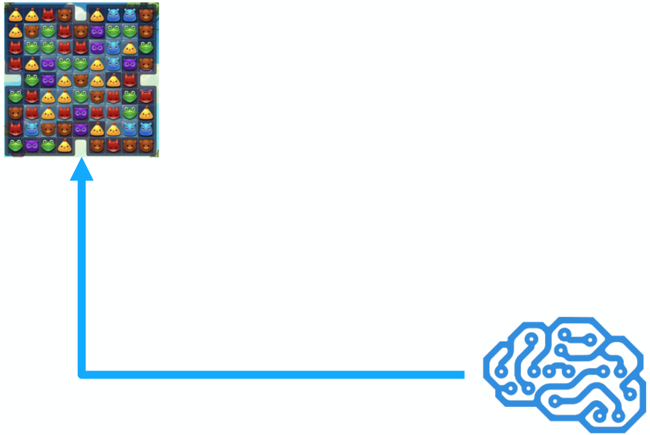
Personalized gameplay
for **each** user



machine learning

Problem Formulation

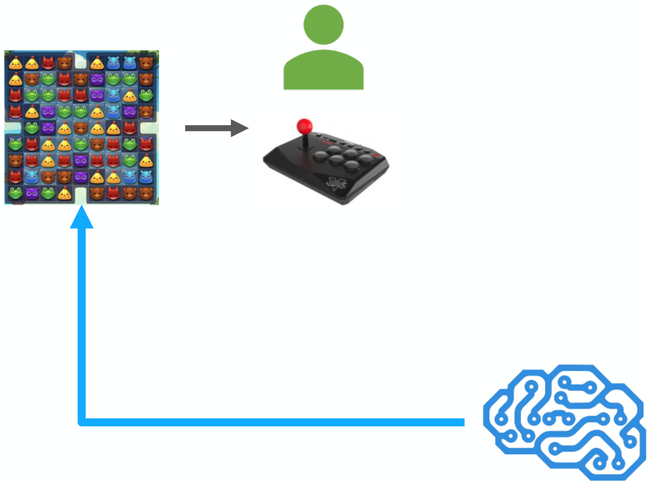
Gameplay modification: Action sequences



Objective: Rewards = player revenue/retention

Problem Formulation

Gameplay modification: Action sequences

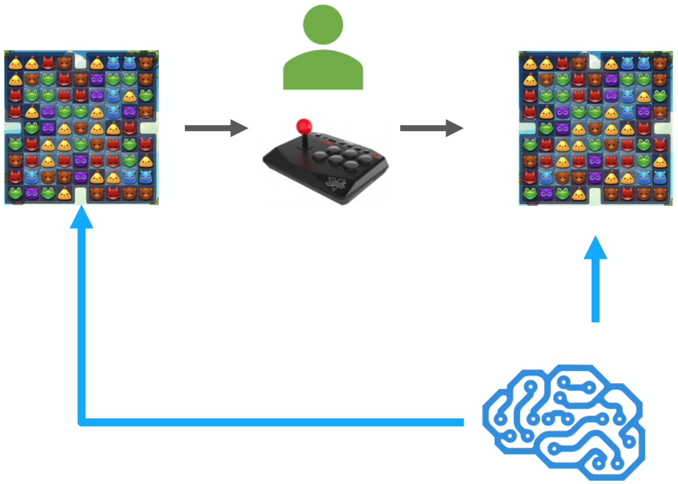


Objective: Rewards = player revenue/retention

Happy Elements

Problem Formulation

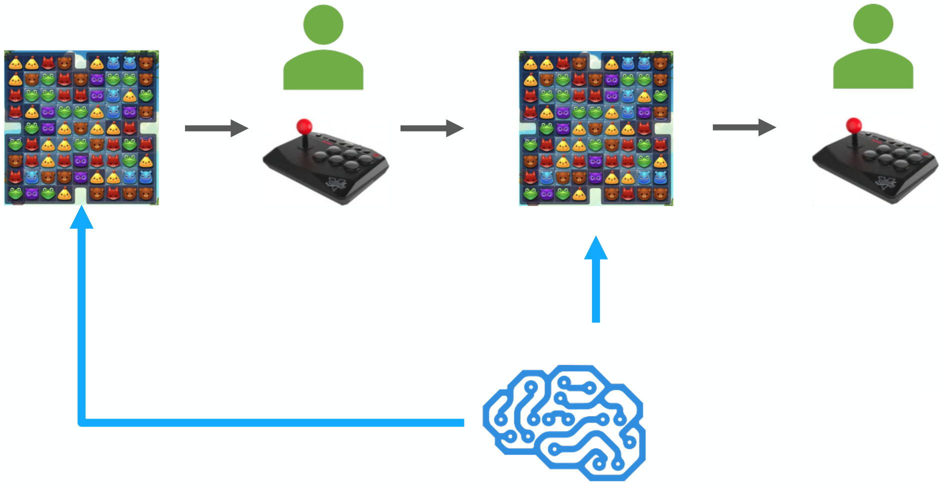
Gameplay modification: Action sequences



Objective: Rewards = player revenue/retention

Problem Formulation

Gameplay modification: Action sequences

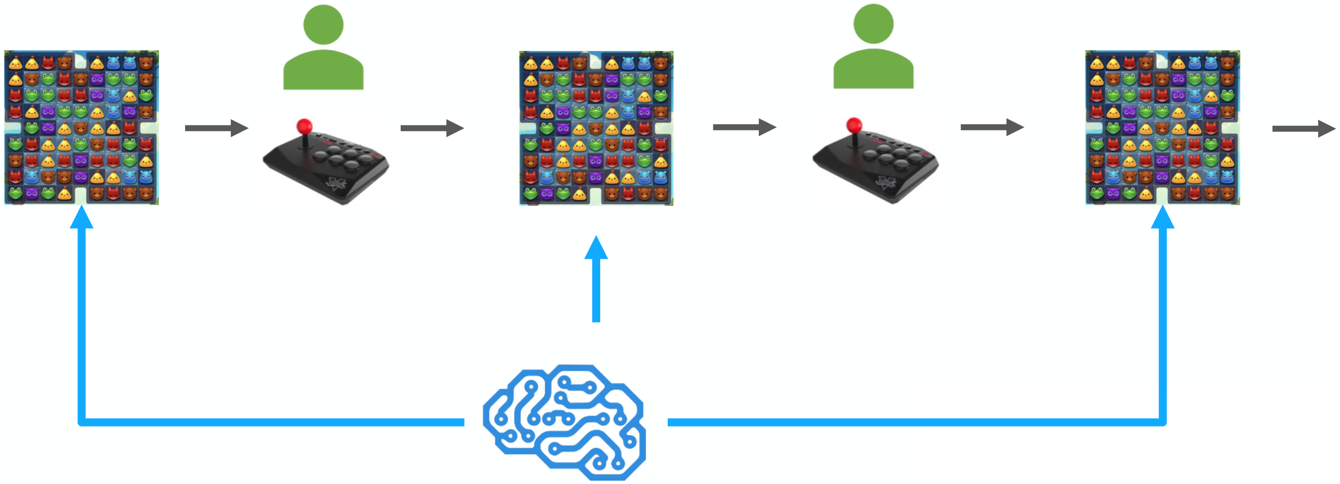


Objective: Rewards = player revenue/retention

Happy Elements

Problem Formulation

Gameplay modification: Action sequences



Objective: Rewards = player revenue/retention

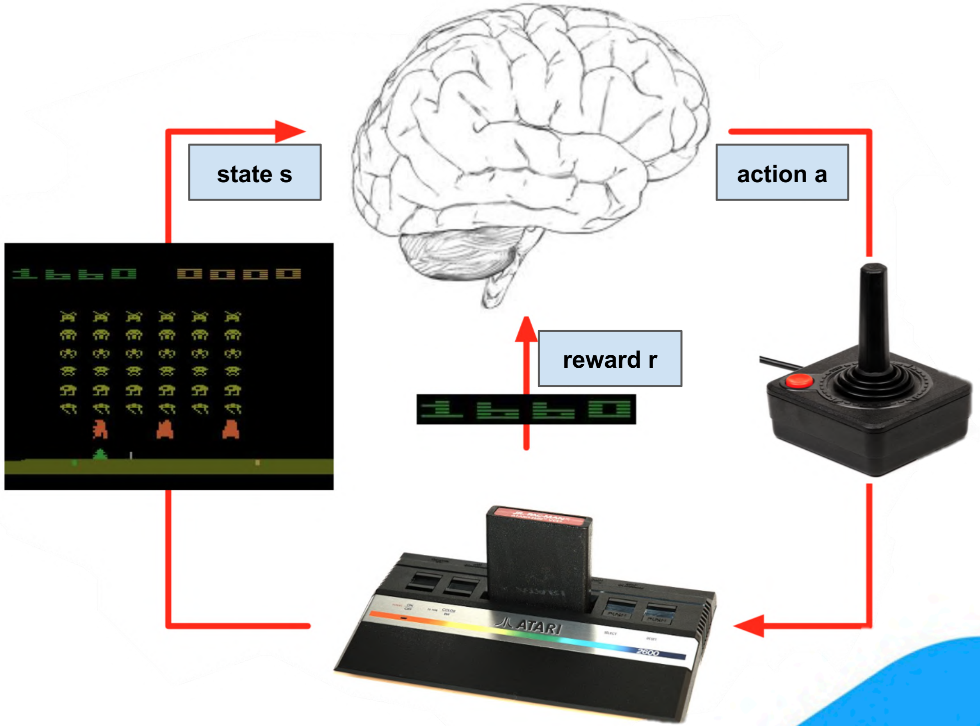
Happy Elements

Problem Formulation

Reinforcement Learning:

- What **action *a*** to take in **state *s*** to **optimize** the expected **reward E[*r*]**?

- For example, video game:
    - state s = screen
    - action a = controller
    - reward r = score



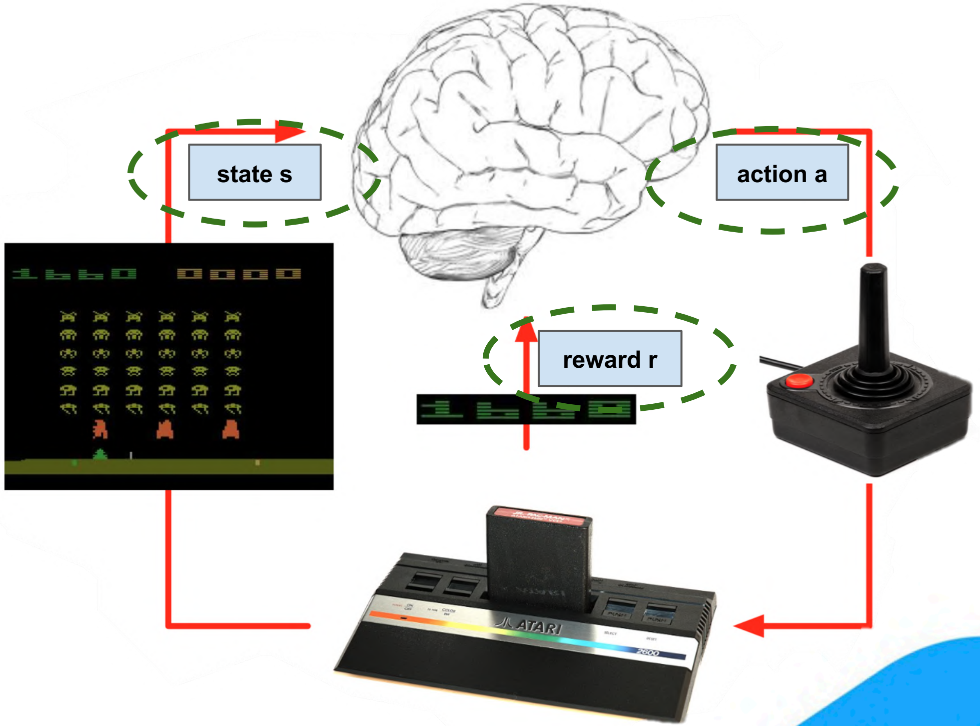
Happy Elements

Problem Formulation

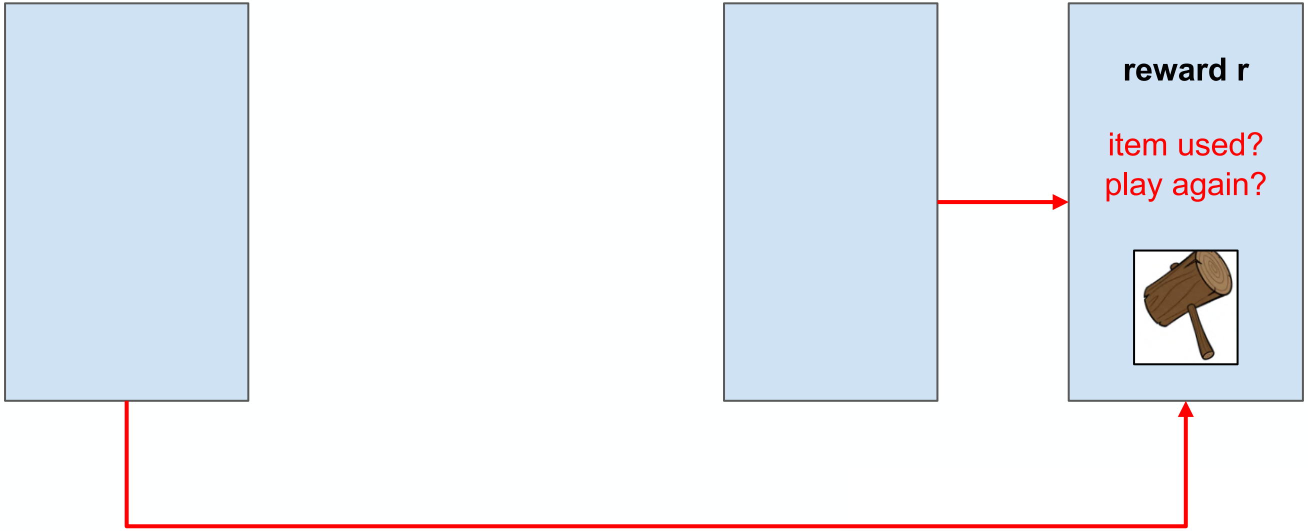
Reinforcement Learning:

- What **action *a*** to take in **state *s*** to **optimize** the expected **reward E[*r*]**?

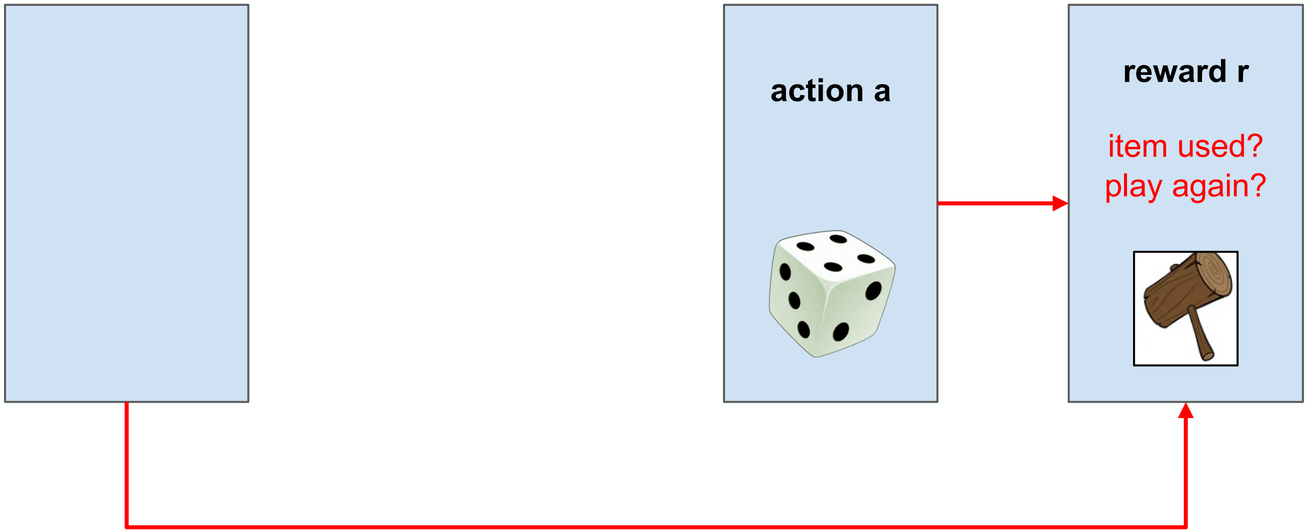
- For example, video game:
    - state s = screen
    - action a = controller
    - reward r = score



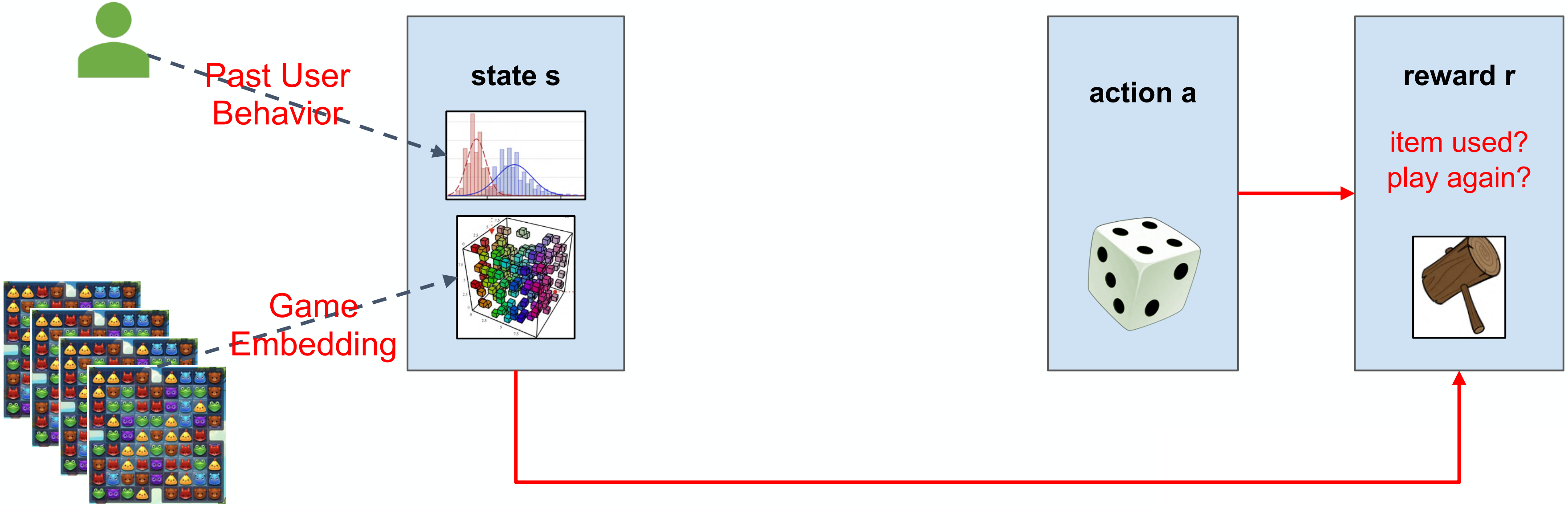
Reinforcement Learning



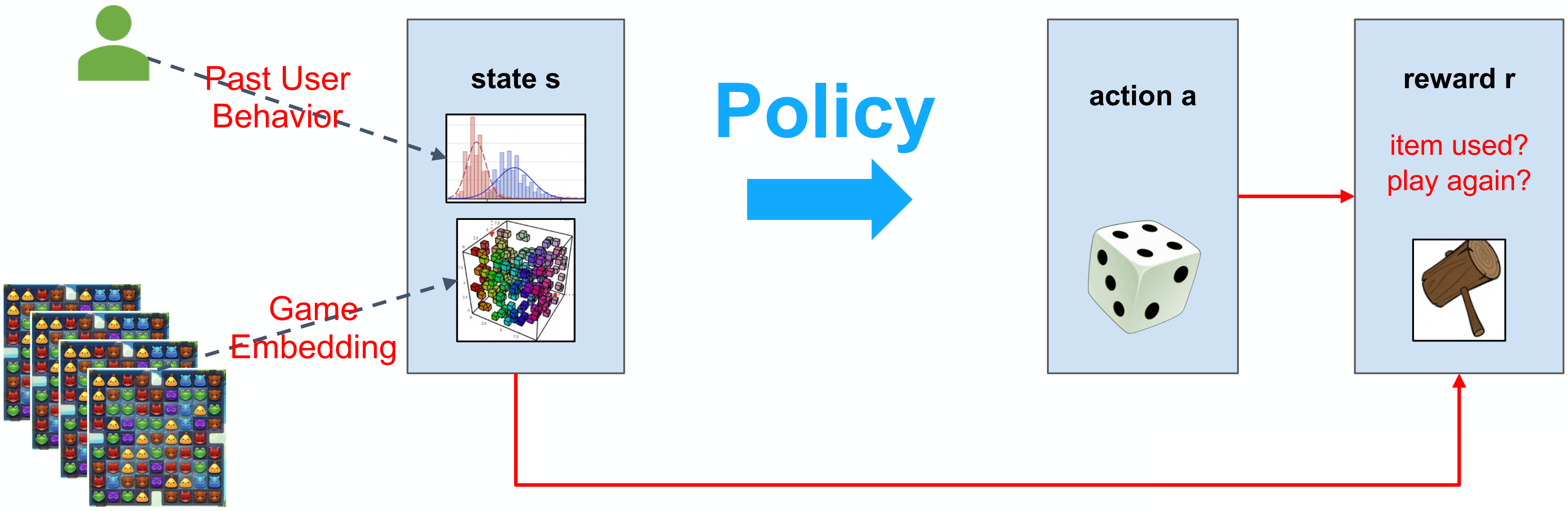
Reinforcement Learning



Reinforcement Learning



Reinforcement Learning

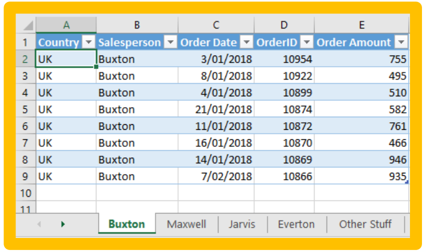


Happy Elements

Deep Learning

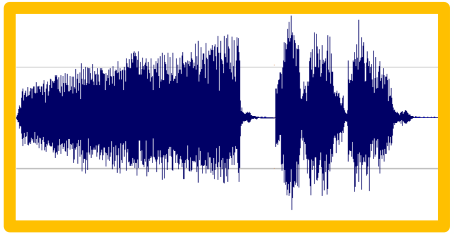
- **Paradigm shift** rather than new technique
- Ability to **optimize any sort** of target using **any type** of data flow
- Extremely **flexible** in fusing and integrating **heterogeneous data**

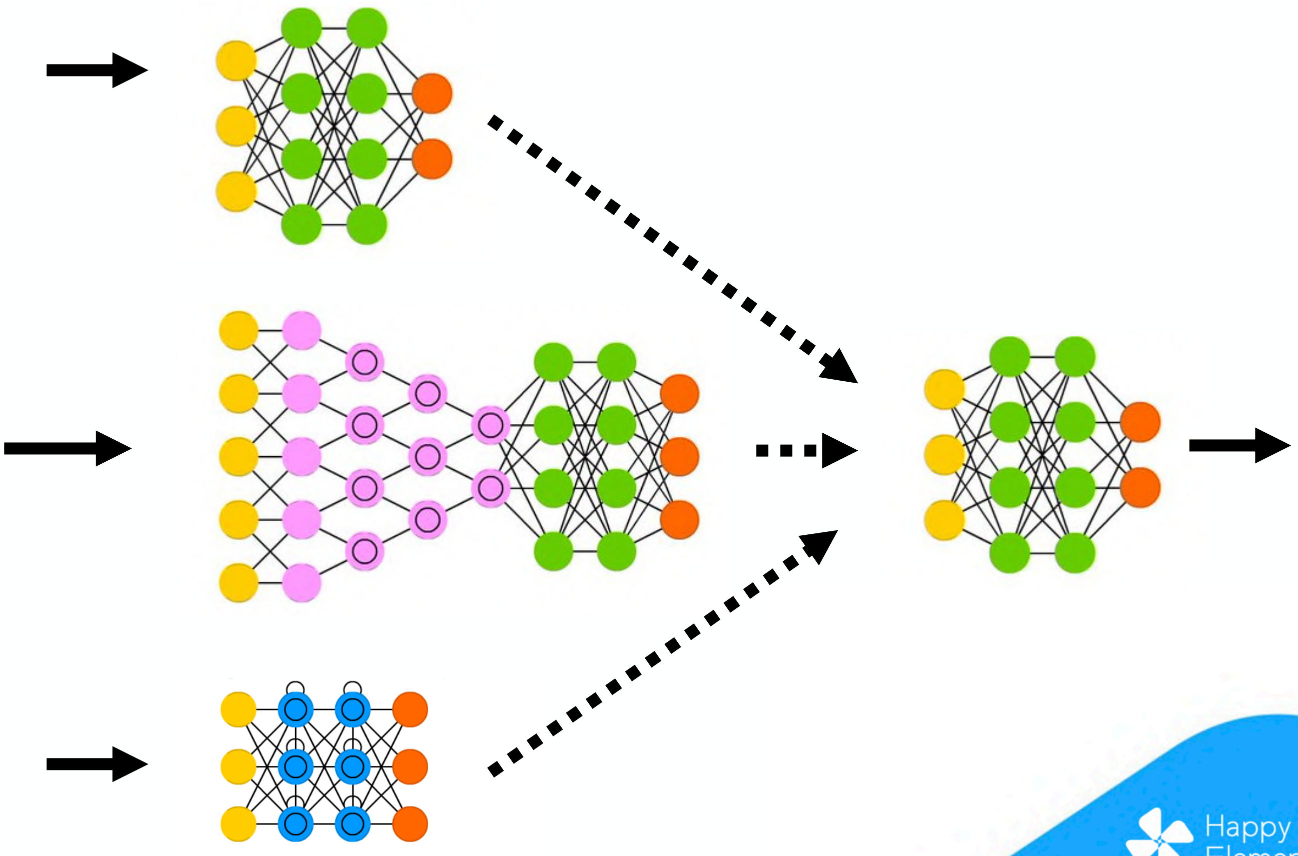
Deep Learning



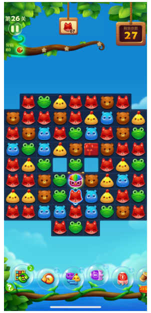
	A	B	C	D	E
1	Country	Salesperson	Order Date	OrderID	Order Amount
2	UK	Buxton	3/01/2018	10954	755
3	UK	Buxton	8/01/2018	10922	495
4	UK	Buxton	4/01/2018	10899	510
5	UK	Buxton	21/01/2018	10874	582
6	UK	Buxton	11/01/2018	10872	761
7	UK	Buxton	16/01/2018	10870	466
8	UK	Buxton	14/01/2018	10869	946
9	UK	Buxton	7/02/2018	10866	935
10					
11					

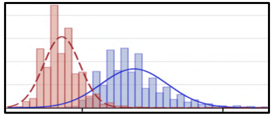
Buxton | Maxwell | Jarvis | Everton | Other Stuff

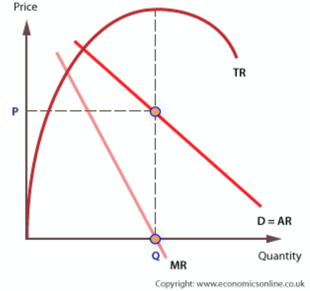


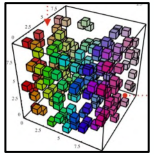


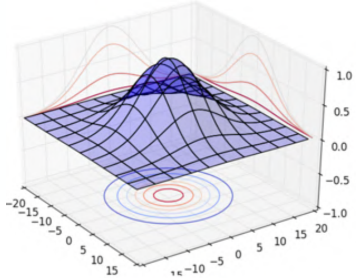
Happy Elements



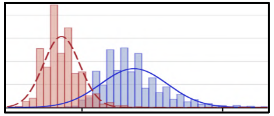


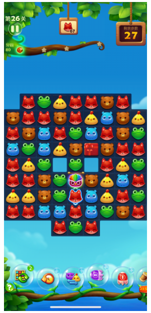


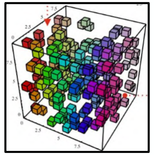


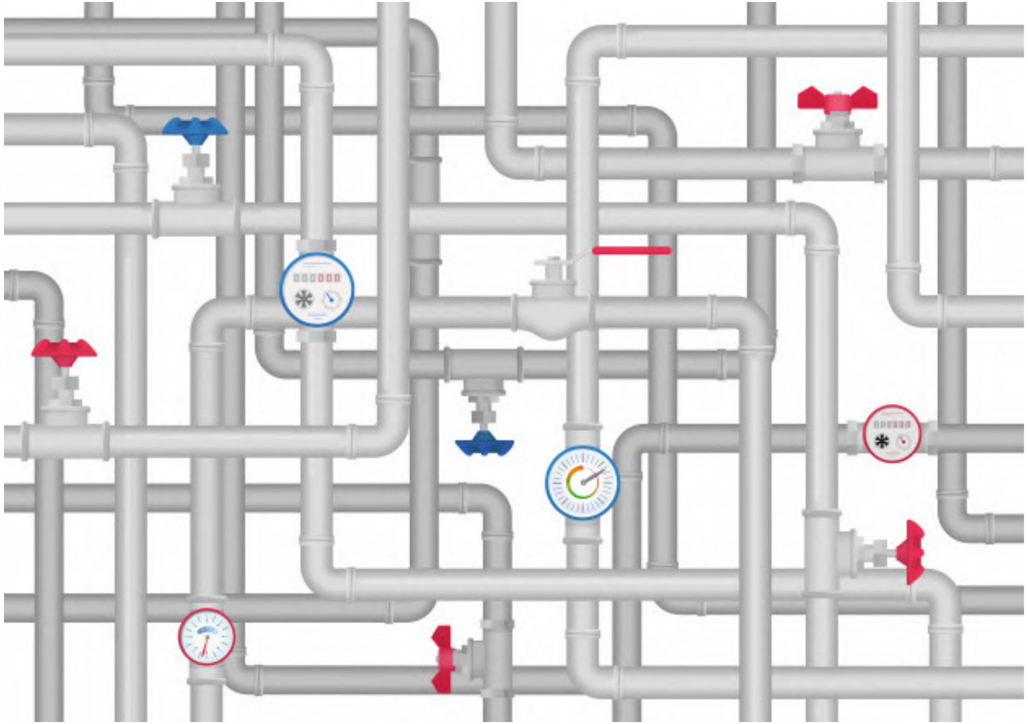


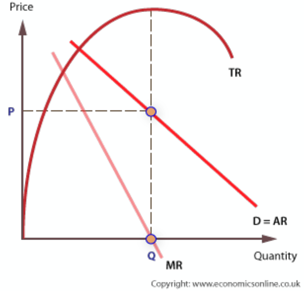
Happy Elements

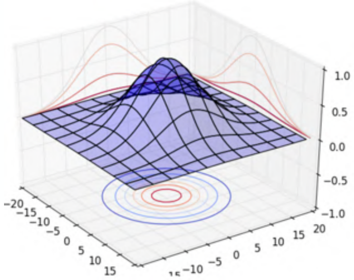




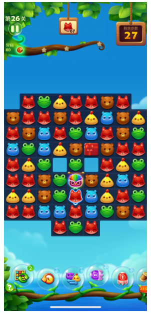


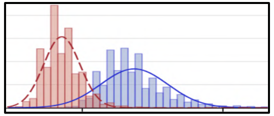


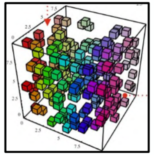


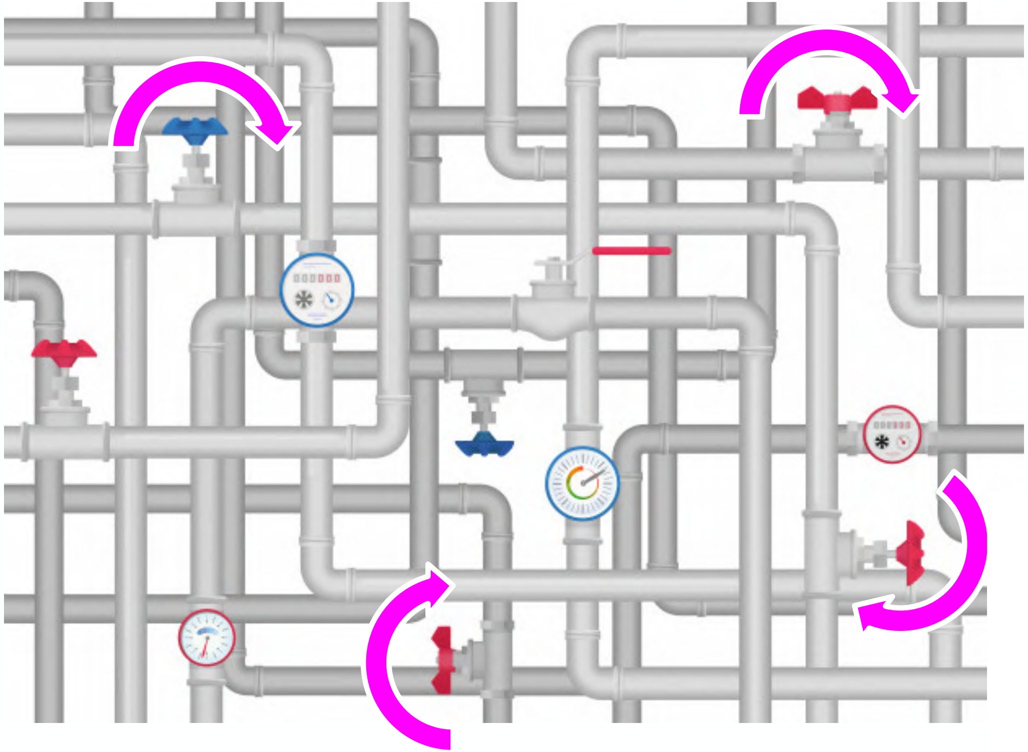


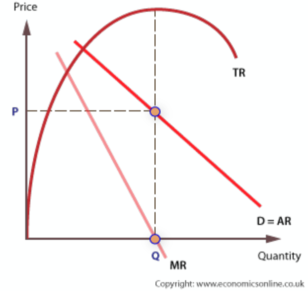
Happy Elements

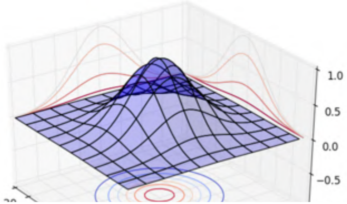






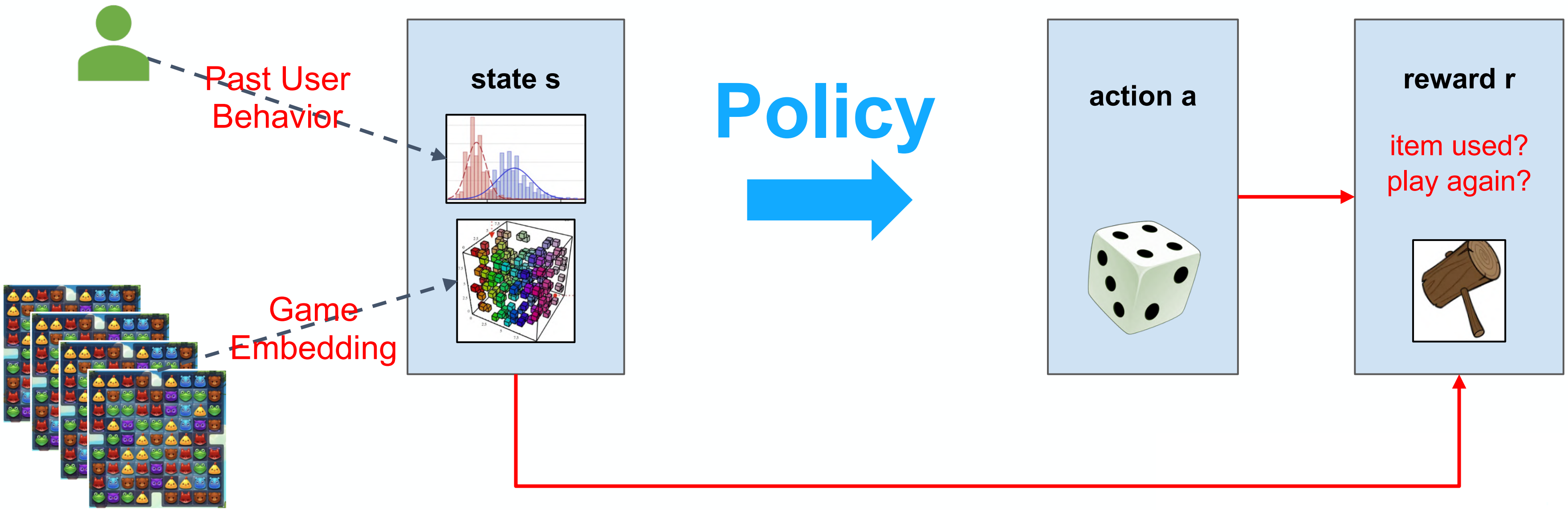






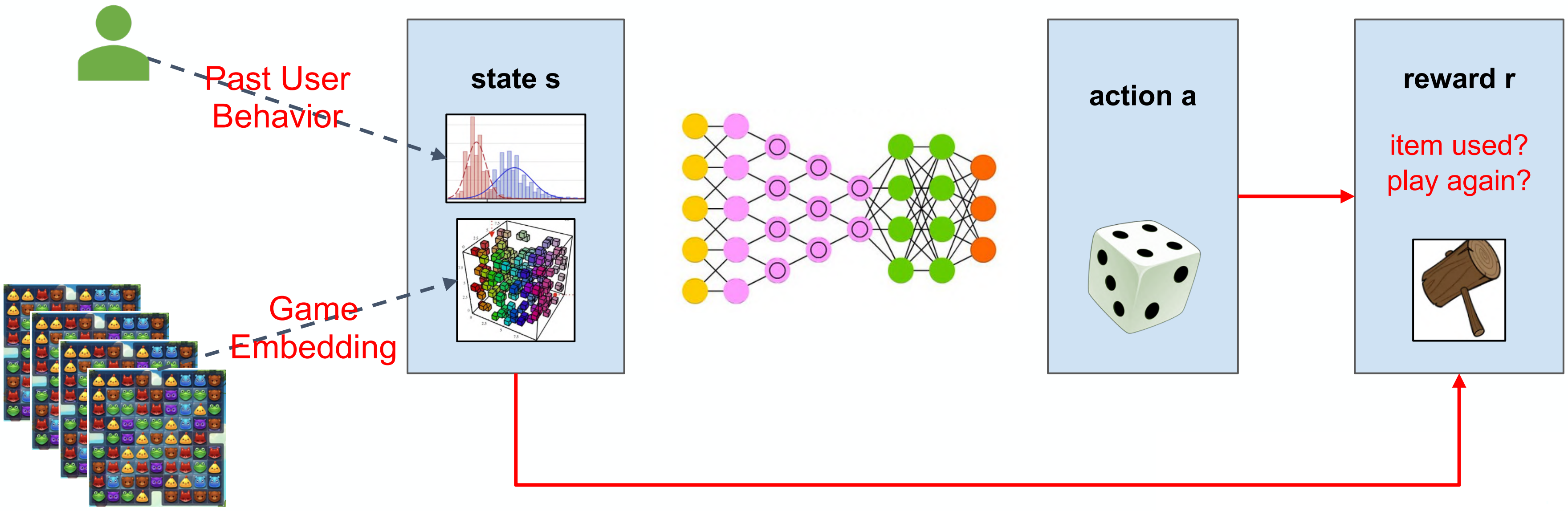
Happy Elements

Reinforcement Learning



Happy Elements

Deep Reinforcement Learning



In Practice: Key Problems

- ML good at optimizing short-term targets:

  How do short-term targets relate to long-term objectives?

- ML good at optimizing on fixed dataset:

  What when the data regime is highly non-stationary?

Optimization Horizon

- Often it is easy to define **short-term targets**:

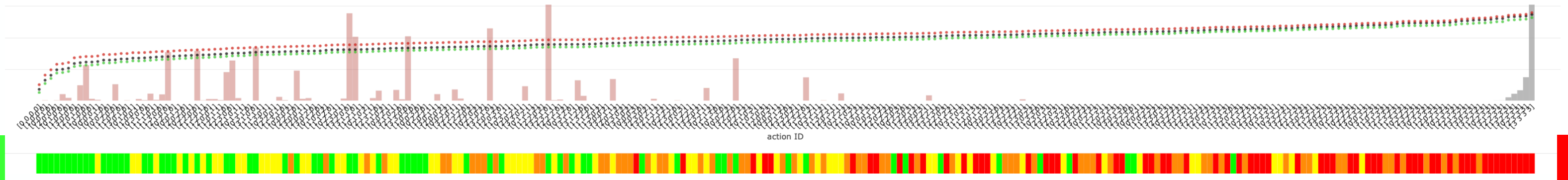
  Did the user play another in-game level?
  Did the user make an in-app purchase?

- But how does this lead to **long-term objectives**?

  User engagement over the next year
  Life-time value of player

Optimization Horizon

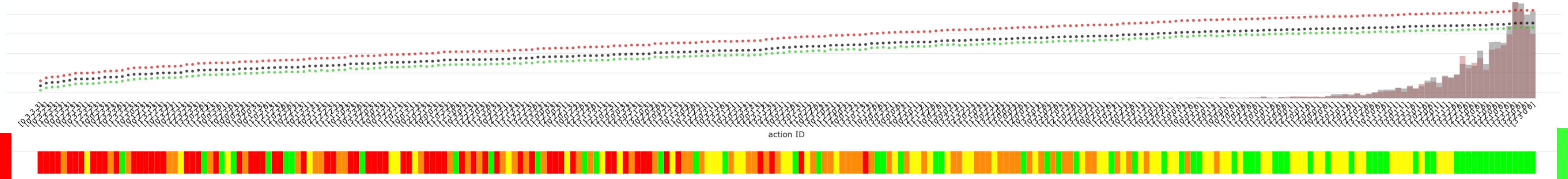
Retention



HARD

EASY

Revenue



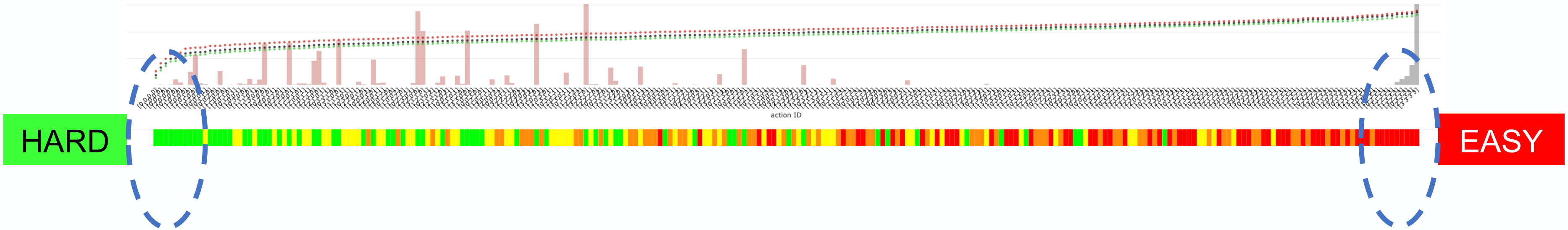
EASY

HARD

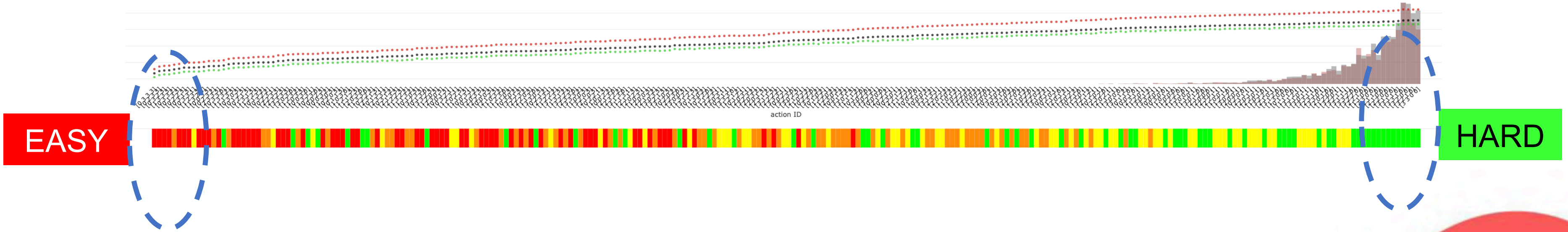
Happy Elements

Optimization Horizon

Retention



Revenue

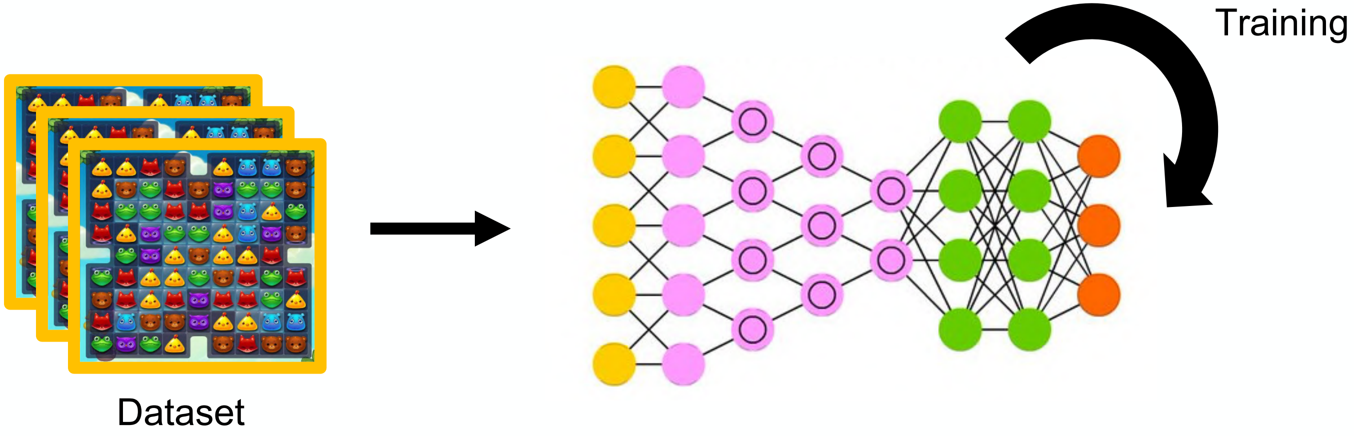


Optimization Horizon

- This can be solved through **RL** by **formulating the right objective**
- Objective is a **sum of individual short-term targets** over a time horizon
- However problem remains in **how to accurate model** this objective

Happy Elements

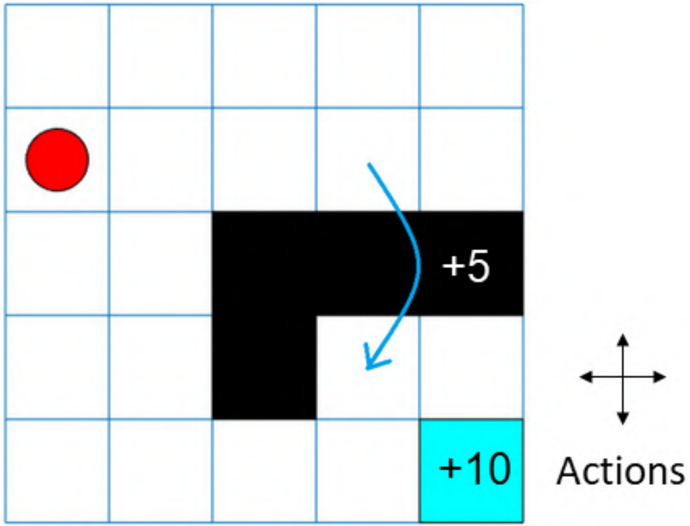
Exploration – Exploitation Duality



- Traditionally, ML works on a **fixed dataset**
- Practical RL in **constant motion**: model generates

Exploration – Exploitation Duality

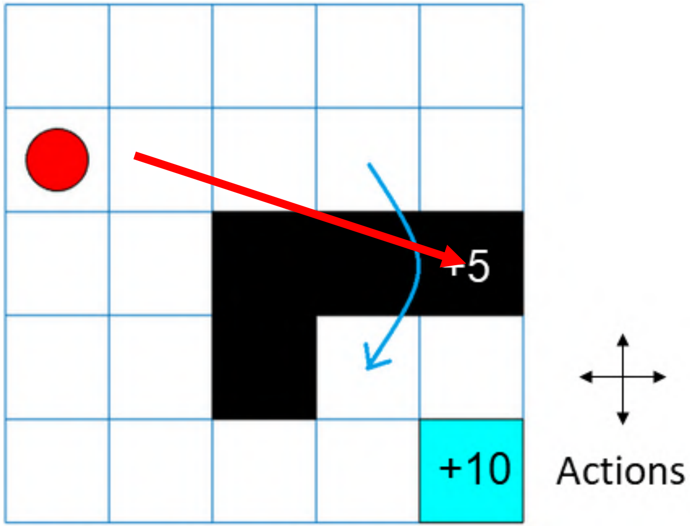
- Rewards might be **sparse**: learn from long-term signal
- + **dynamic interaction** with players: inherently **nonstationary** data regime
- Core problem: trading off **exploration vs exploitation**



Happy Elements

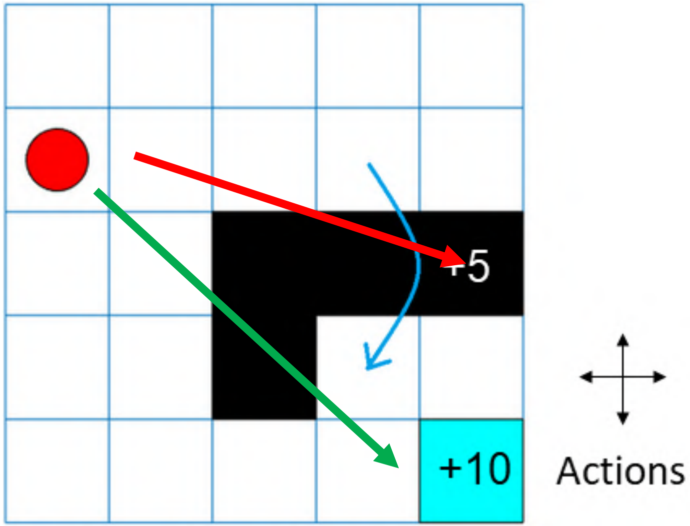
Exploration – Exploitation Duality

- Rewards might be **sparse**: learn from long-term signal
- \+ **dynamic interaction** with players: inherently **nonstationary** data regime
- Core problem: trading off **exploration vs exploitation**



Exploration – Exploitation Duality

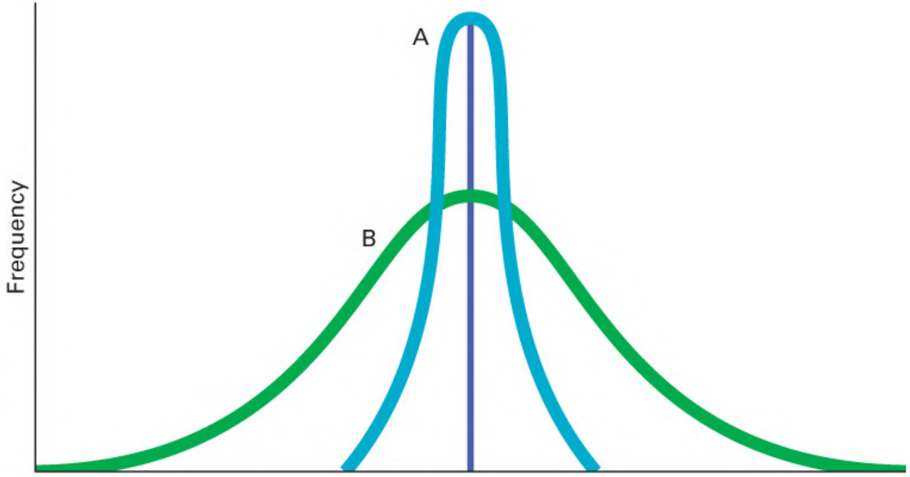
- Rewards might be **sparse**: learn from long-term signal
- + **dynamic interaction** with players: inherently **nonstationary** data regime
- Core problem: trading off **exploration vs exploitation**



Happy Elements

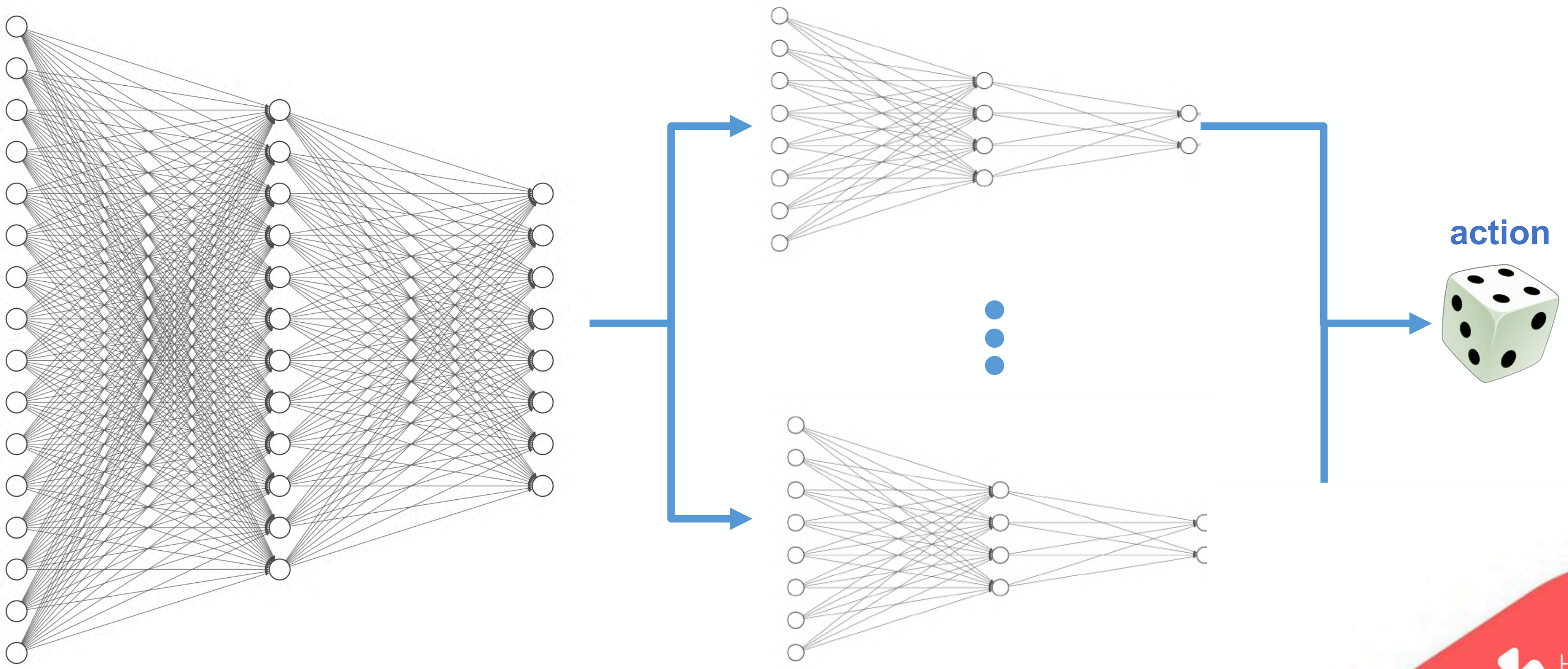
Exploration – Exploitation Duality

- Approximate Bayesian approach to capture **model uncertainty**
- Solving exploitation (A) vs. exploration (B) problem.

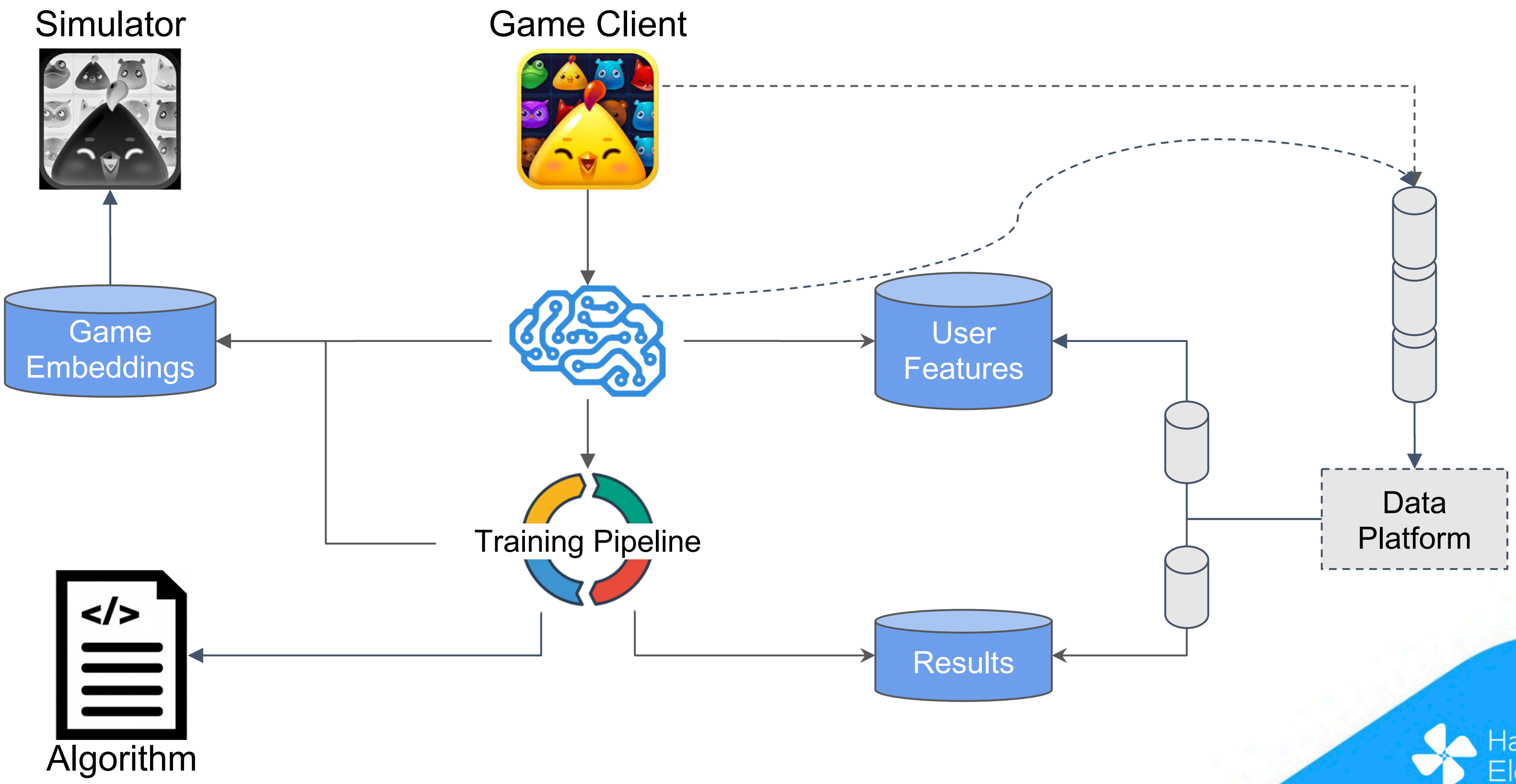


Happy Elements

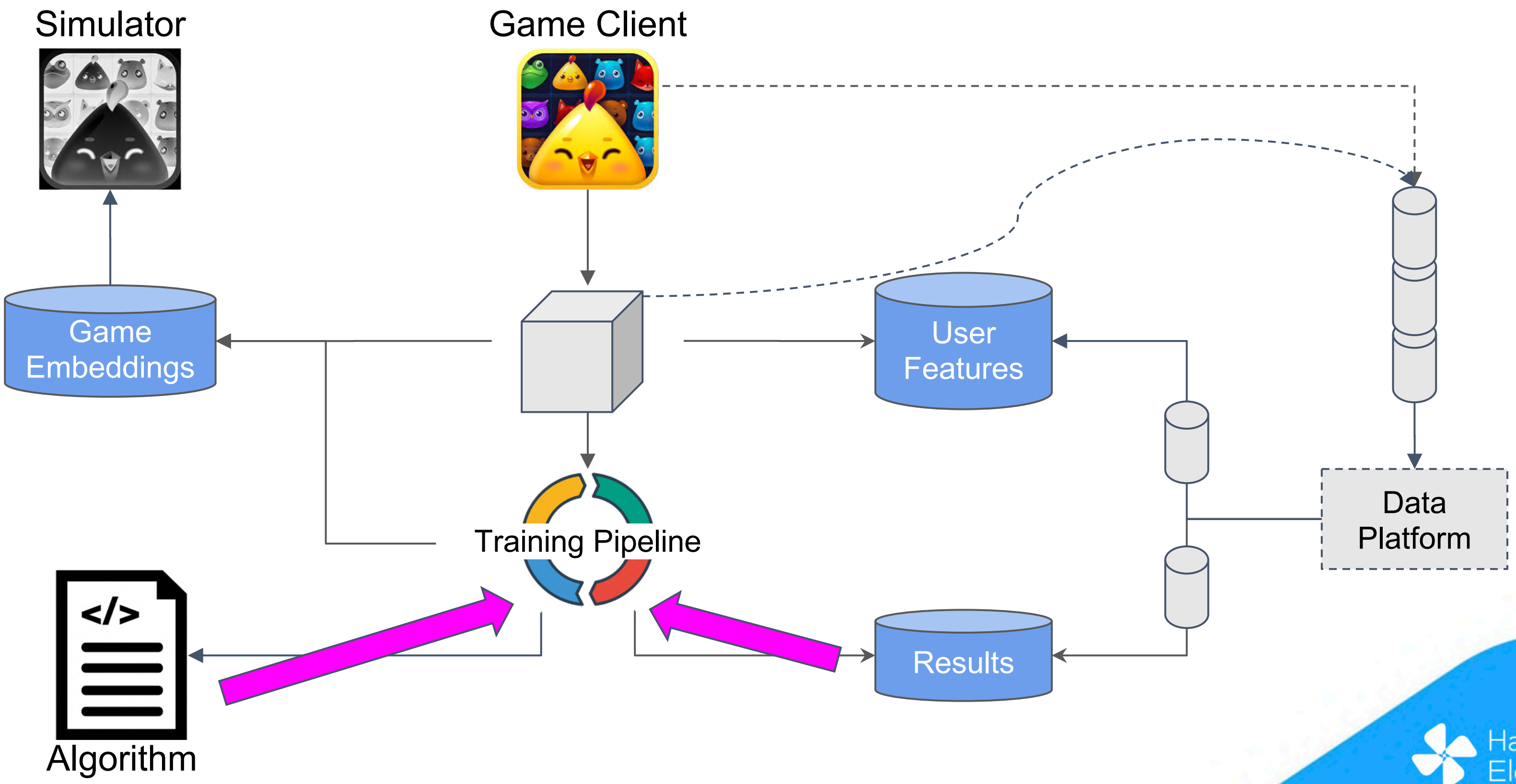
Model: Bootstrapped Contextual Bandits



System Architecture

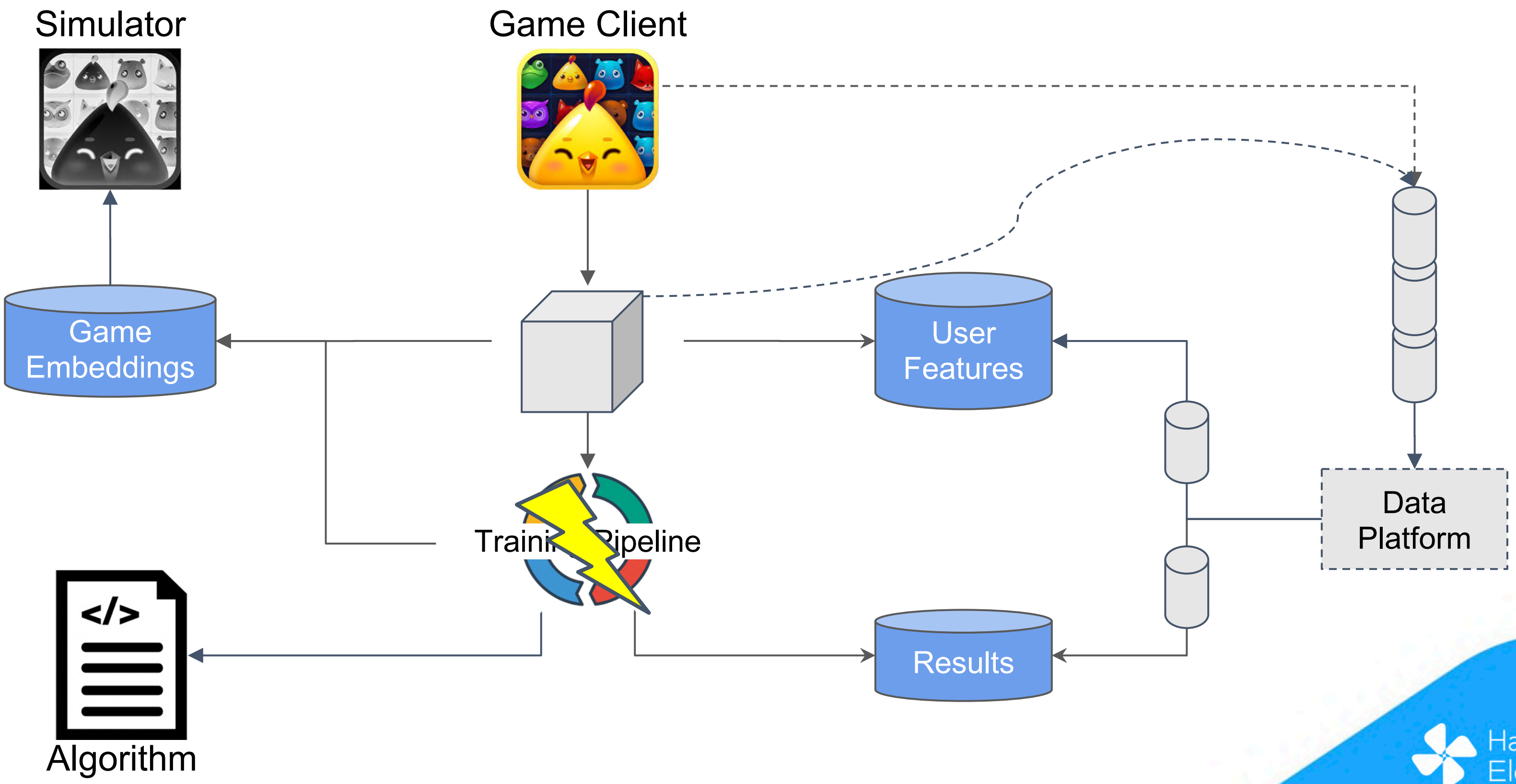


System Architecture



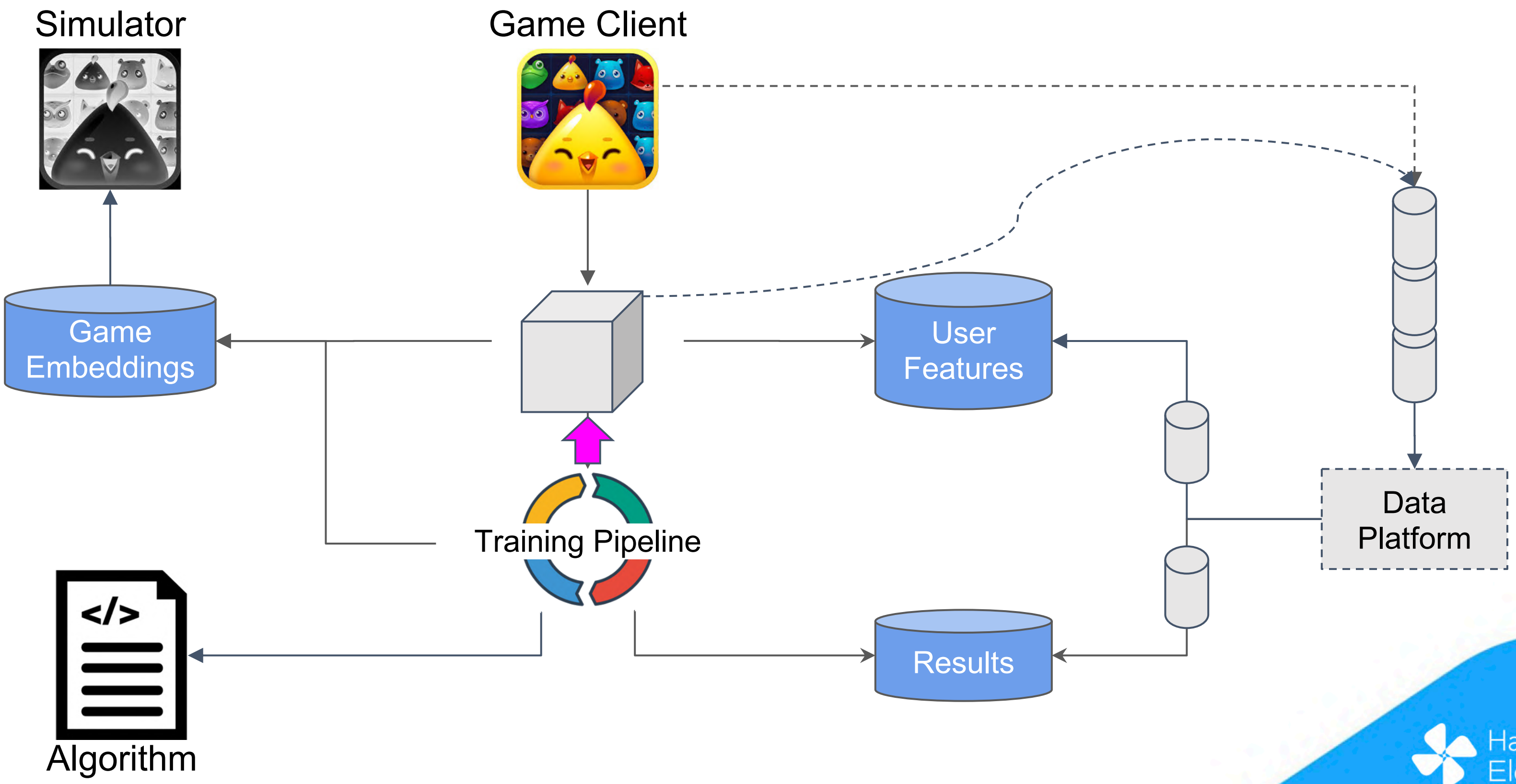
Happy Elements

System Architecture



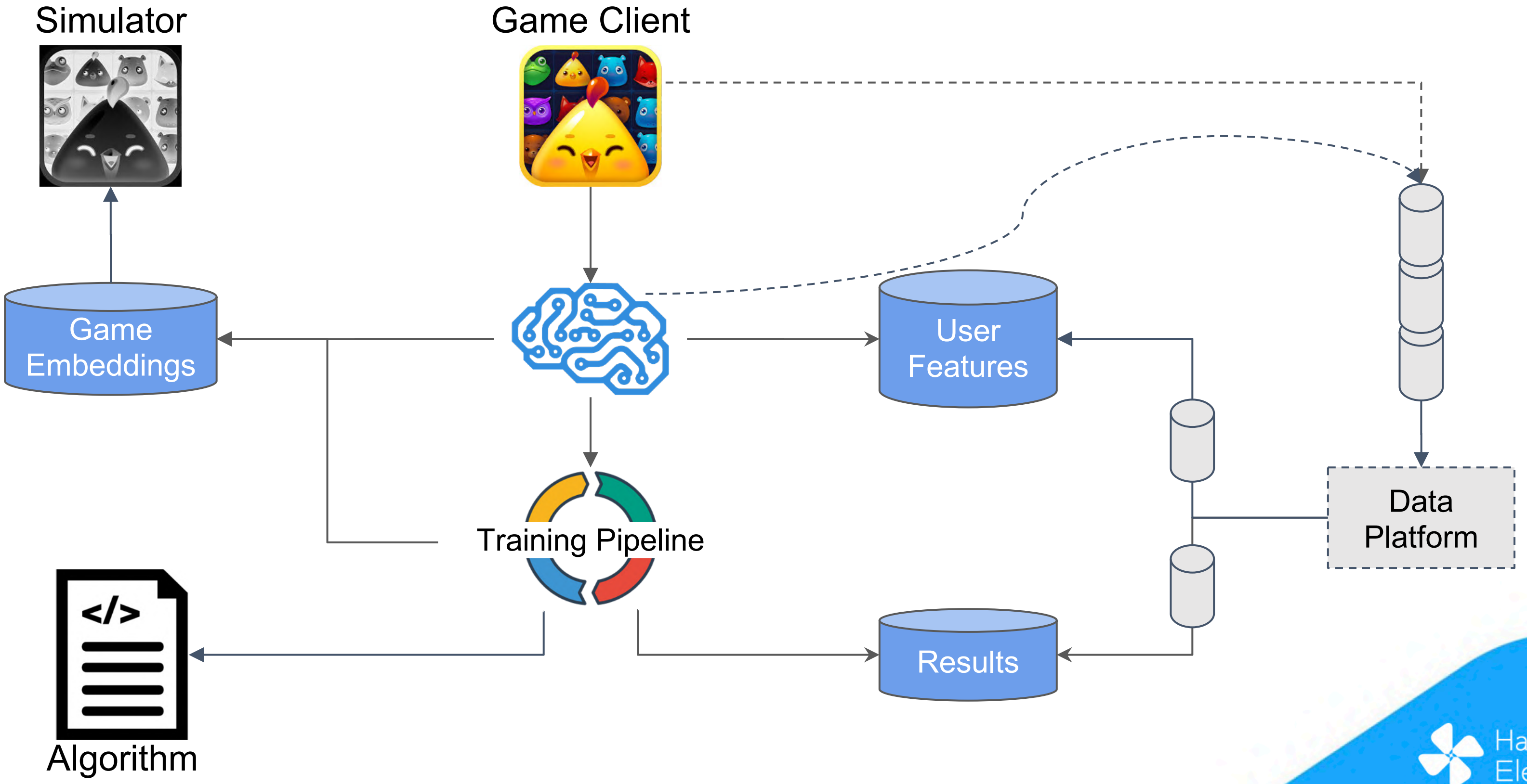
Happy Elements

System Architecture

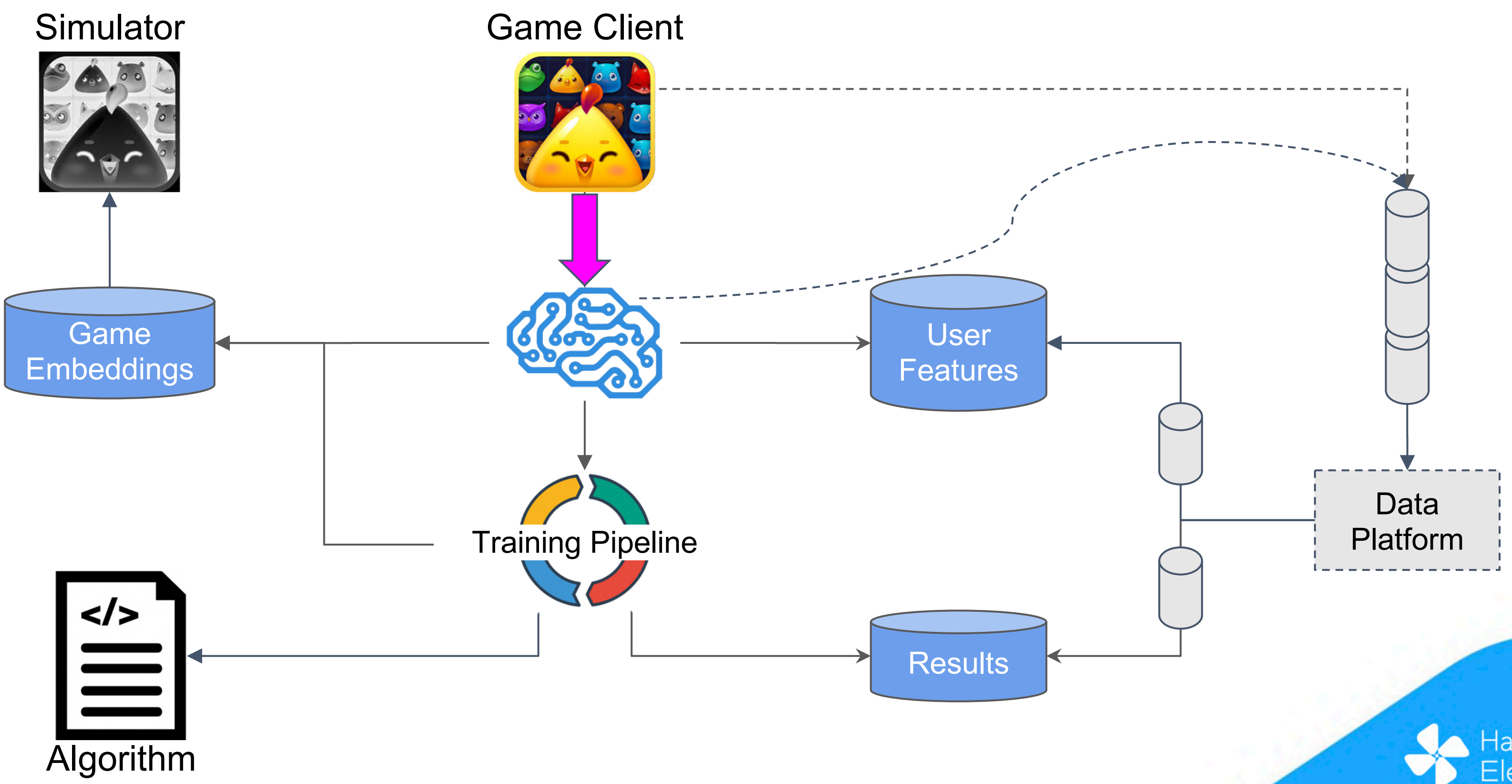


Happy Elements

System Architecture

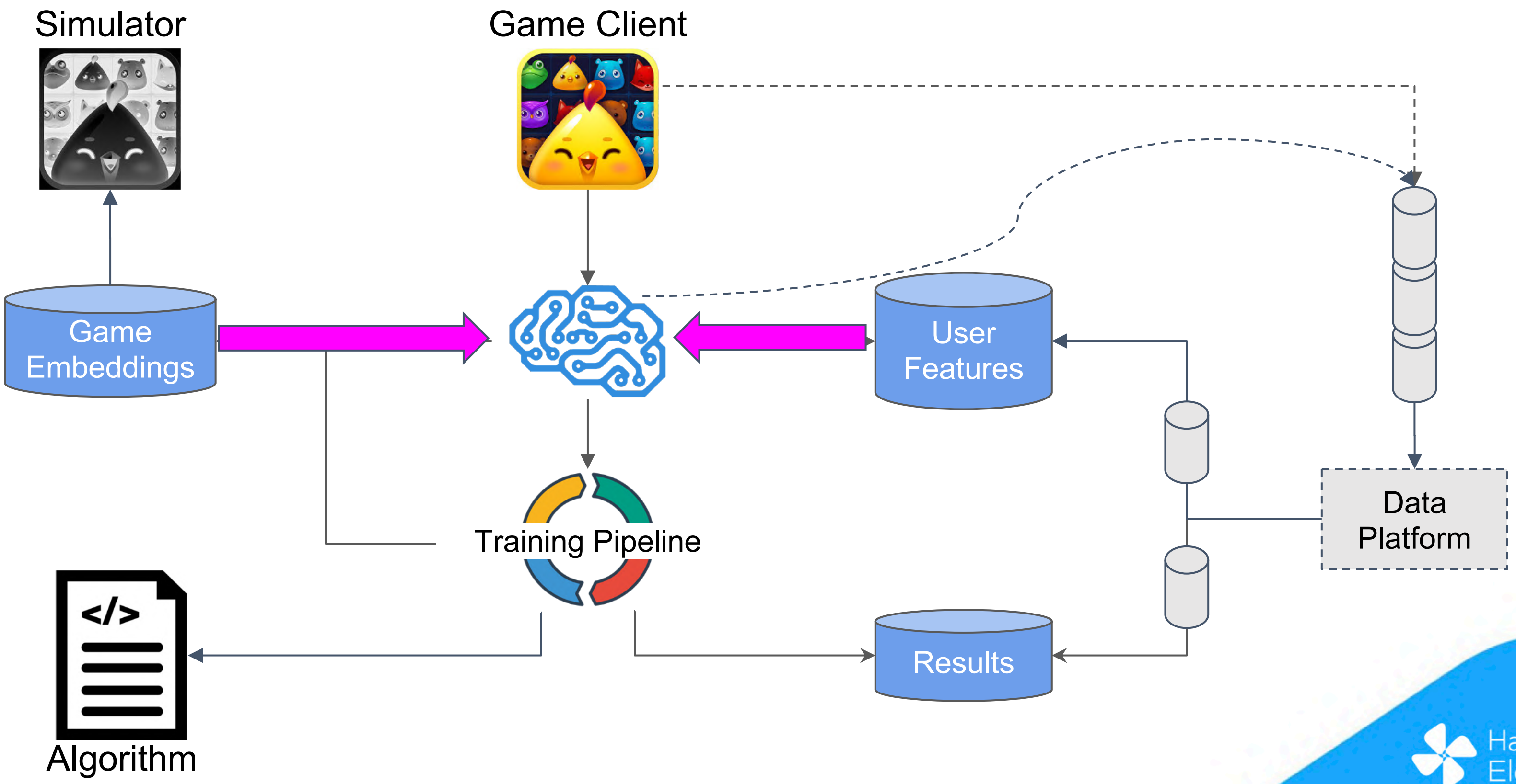


System Architecture

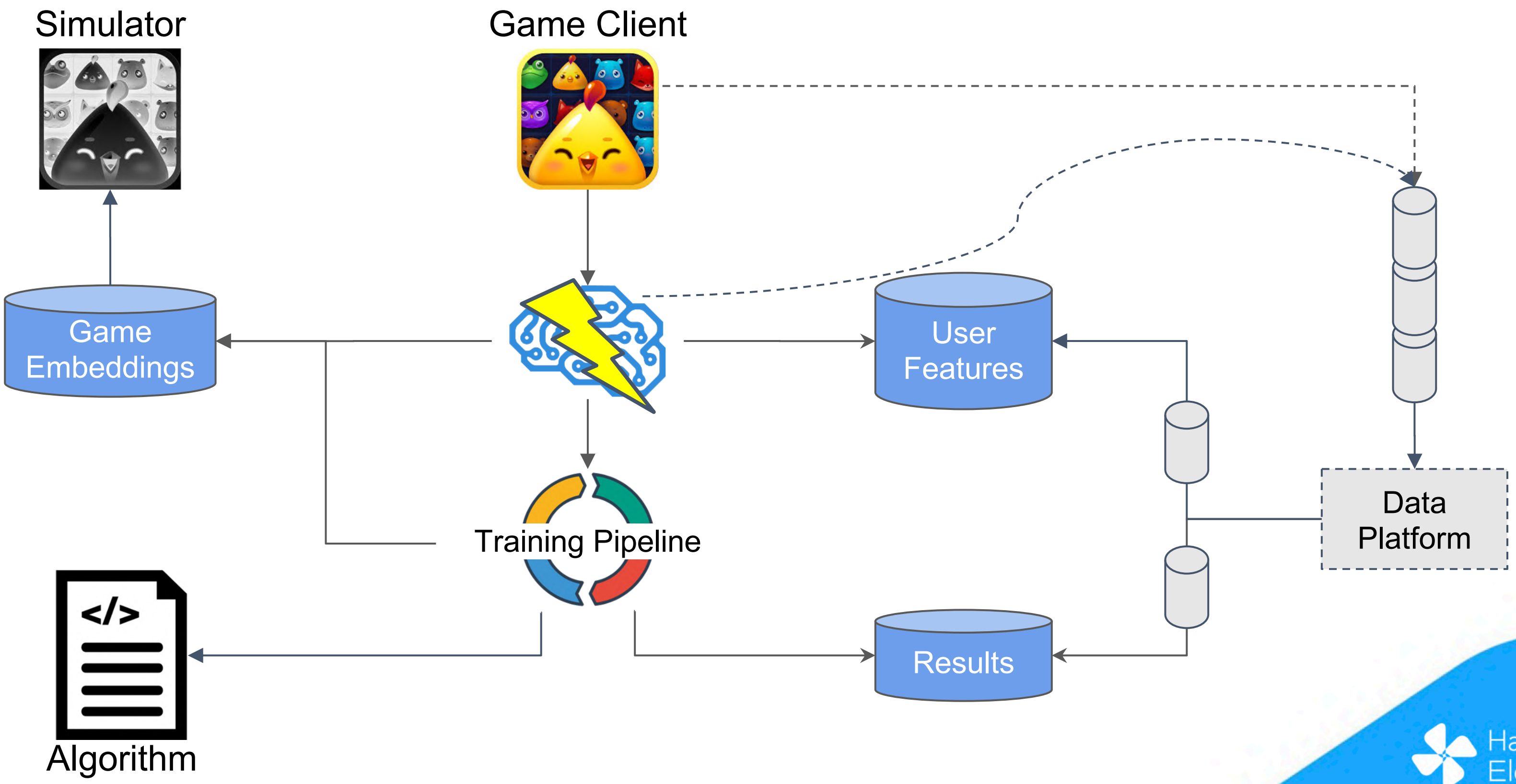


Happy Elements

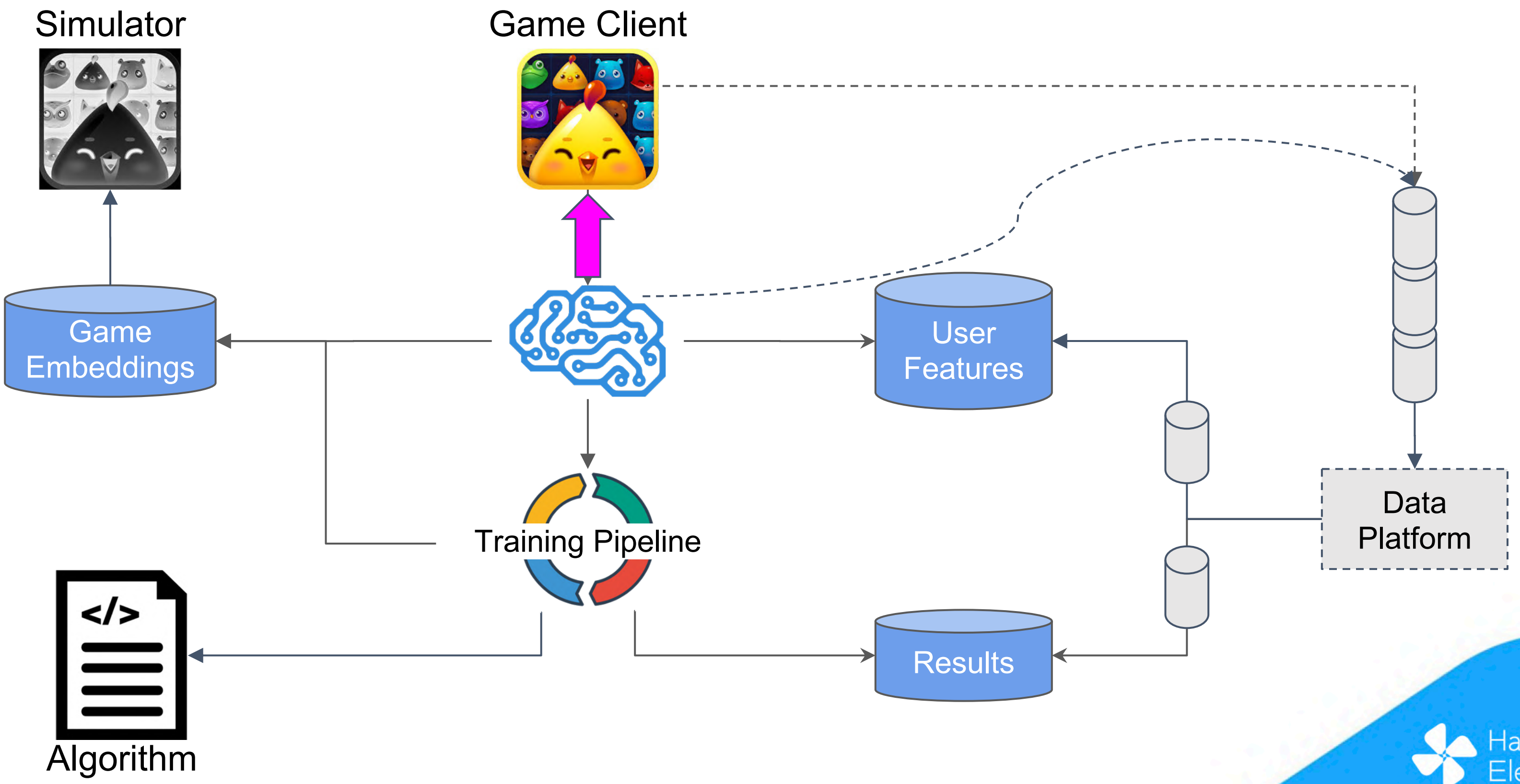
System Architecture



System Architecture

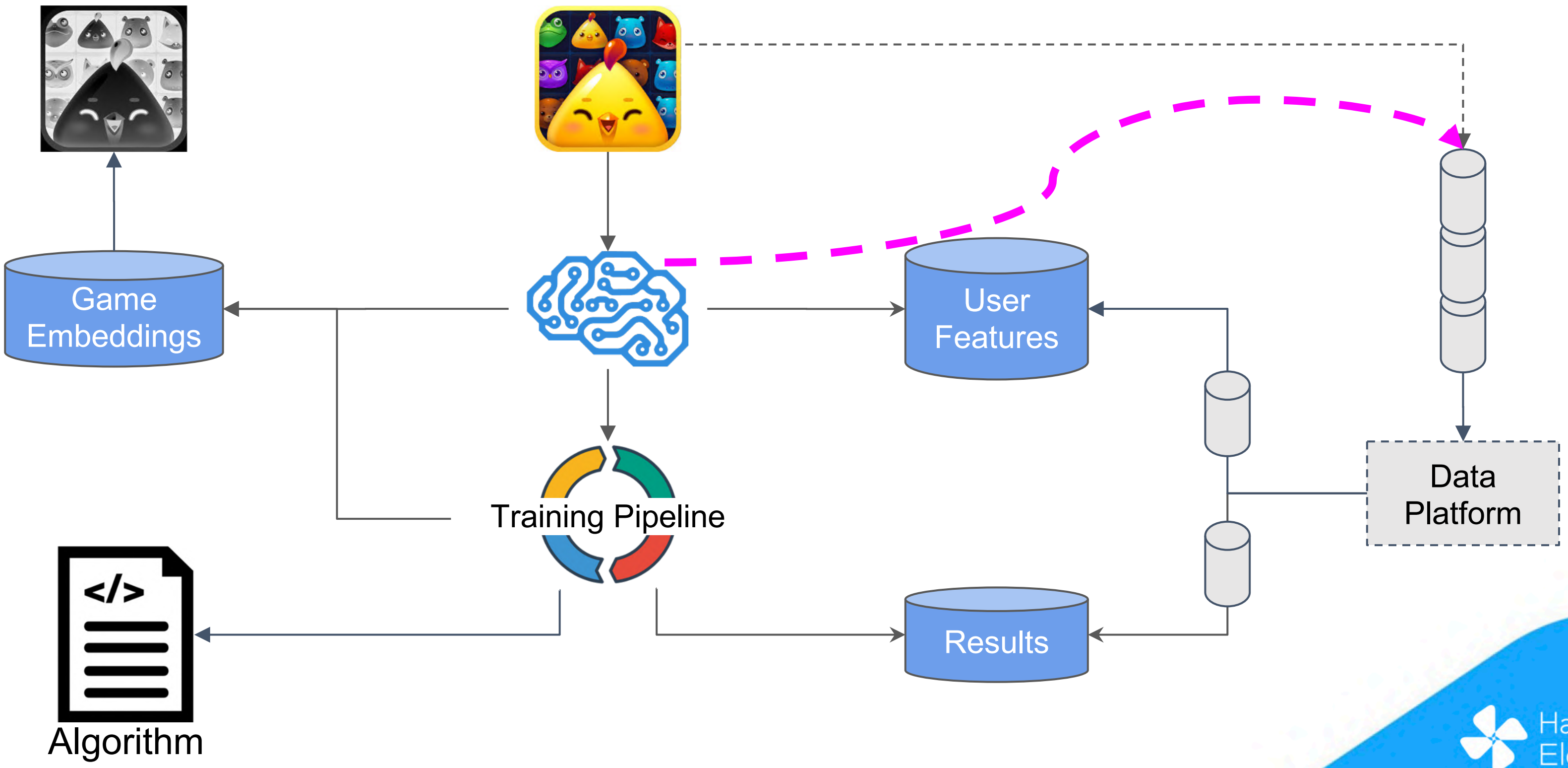


System Architecture



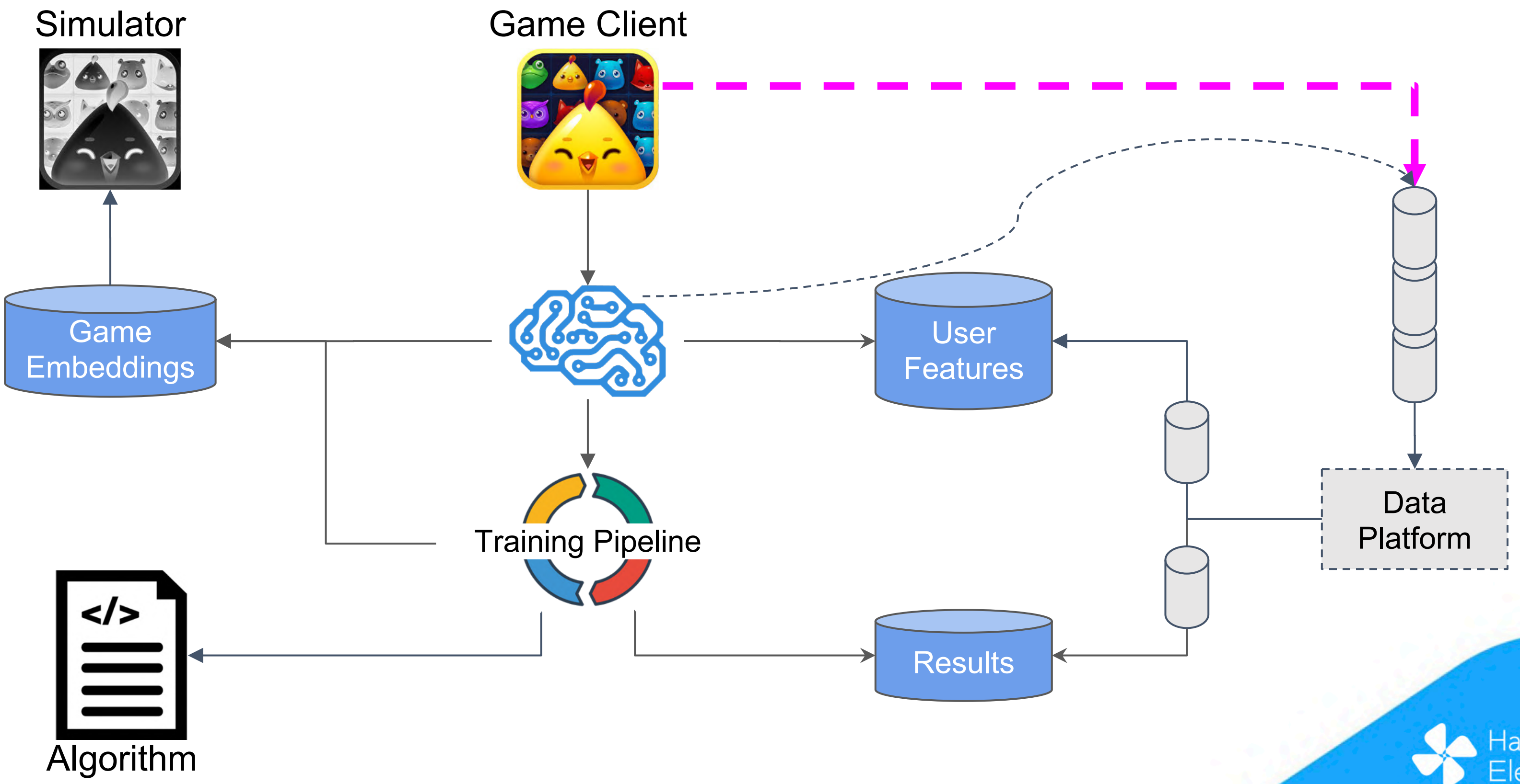
Happy Elements

System Architecture

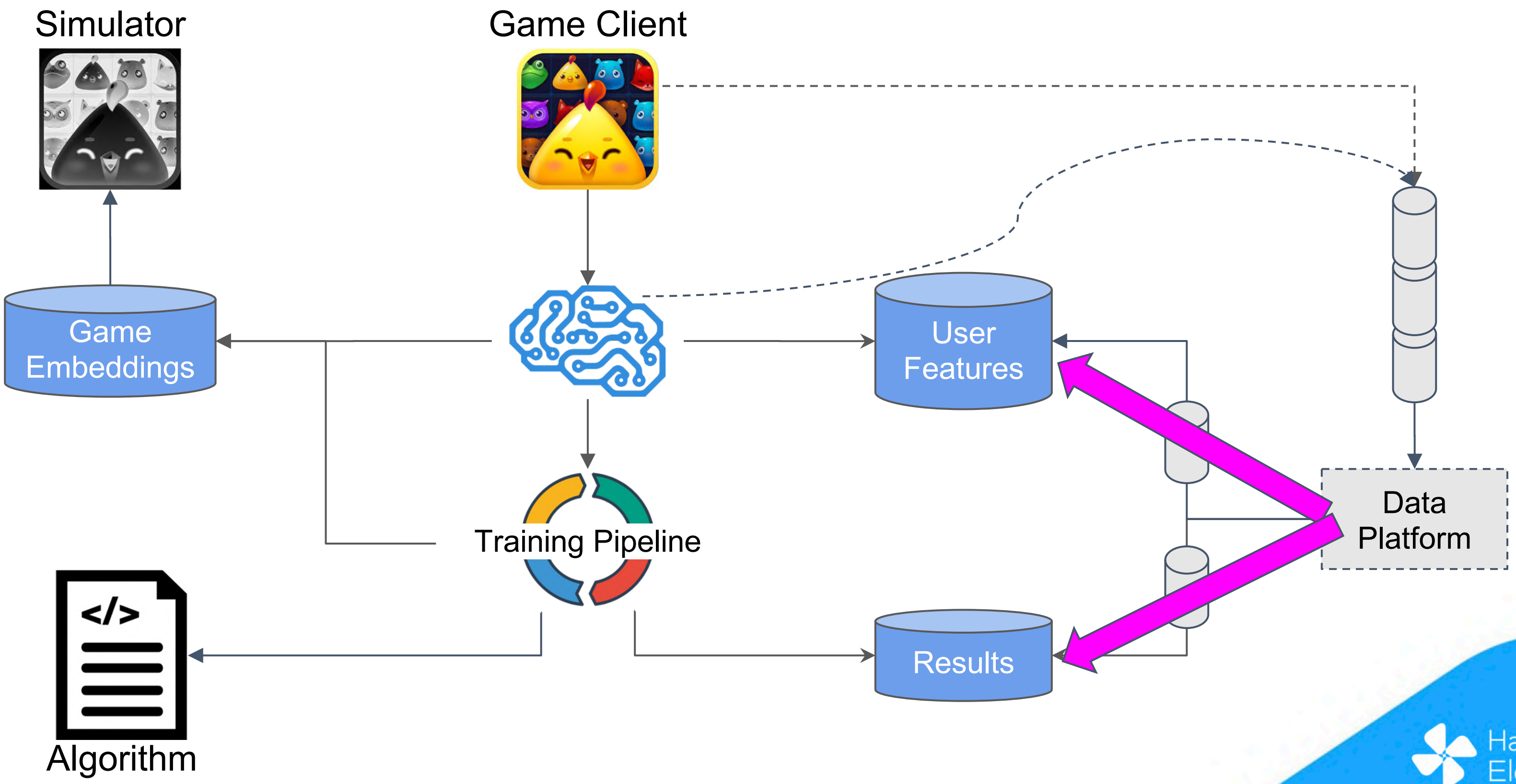


Happy Elements

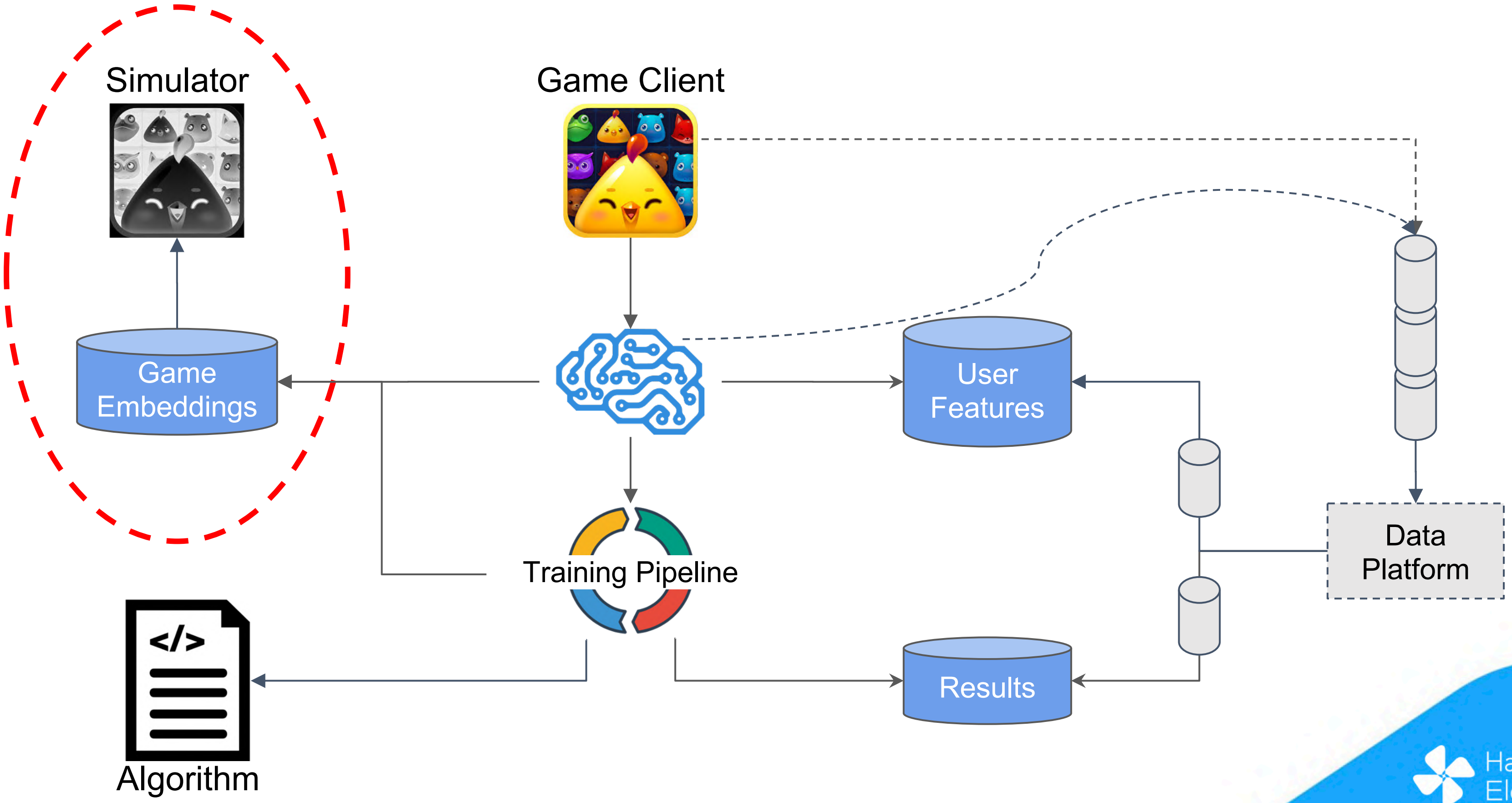
System Architecture



System Architecture

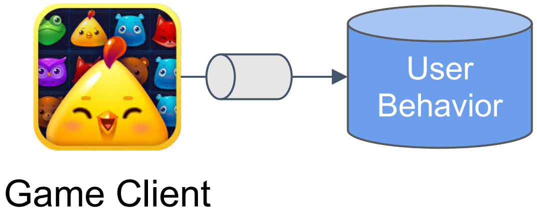


System Architecture

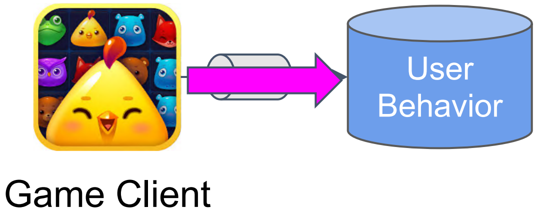


Happy Elements

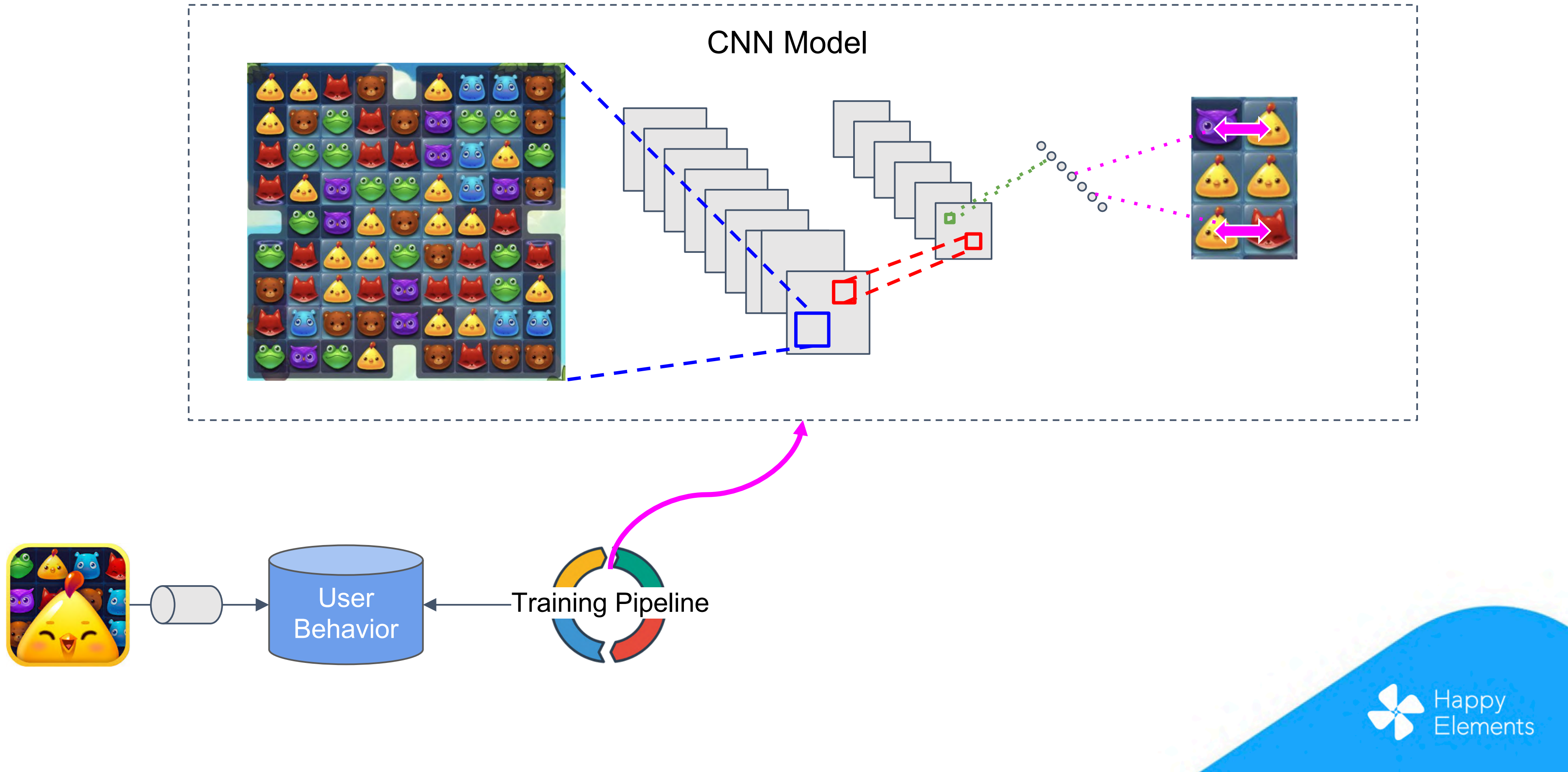
System Architecture: Gameplay Embeddings



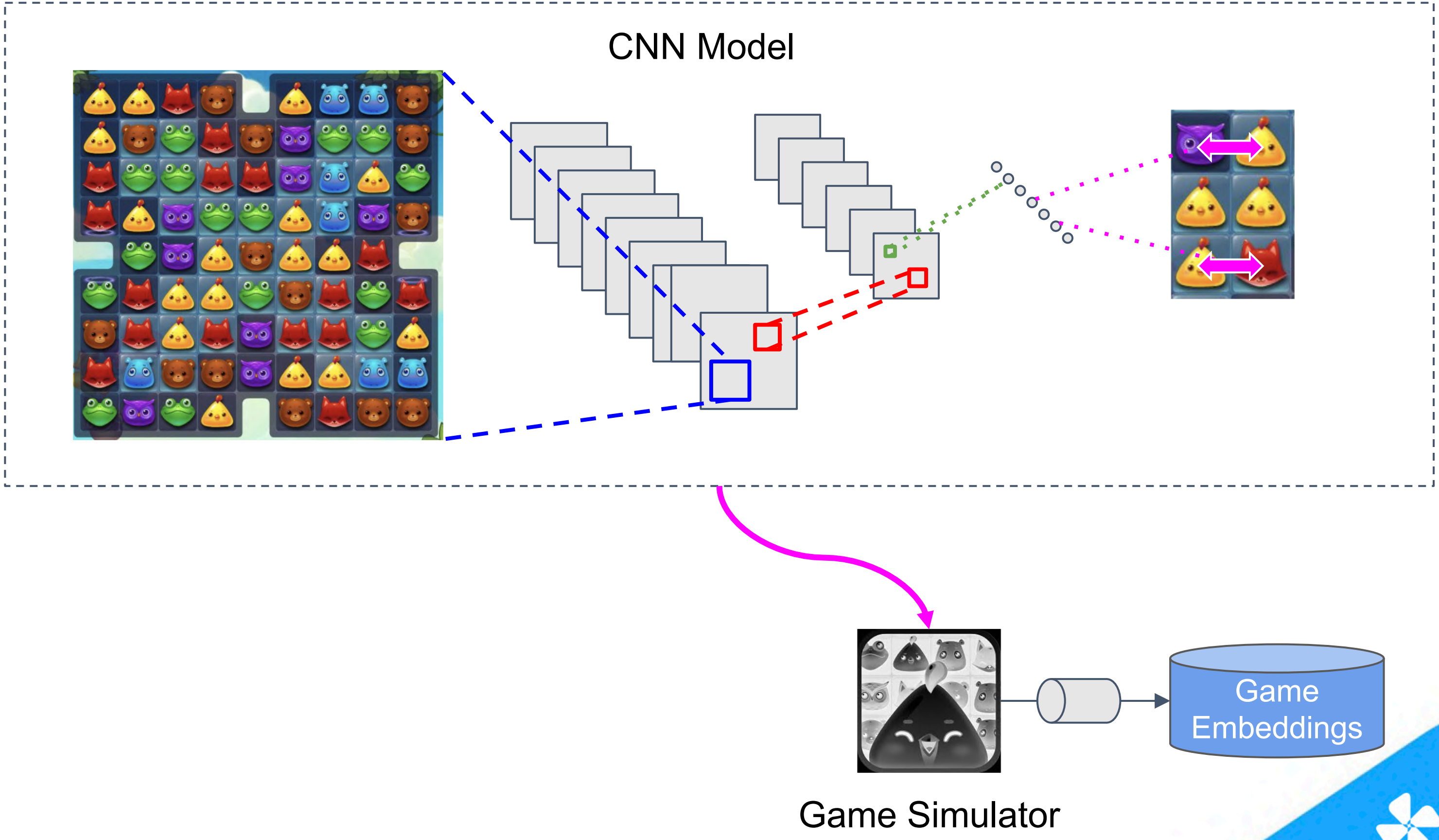
System Architecture: Gameplay Embeddings



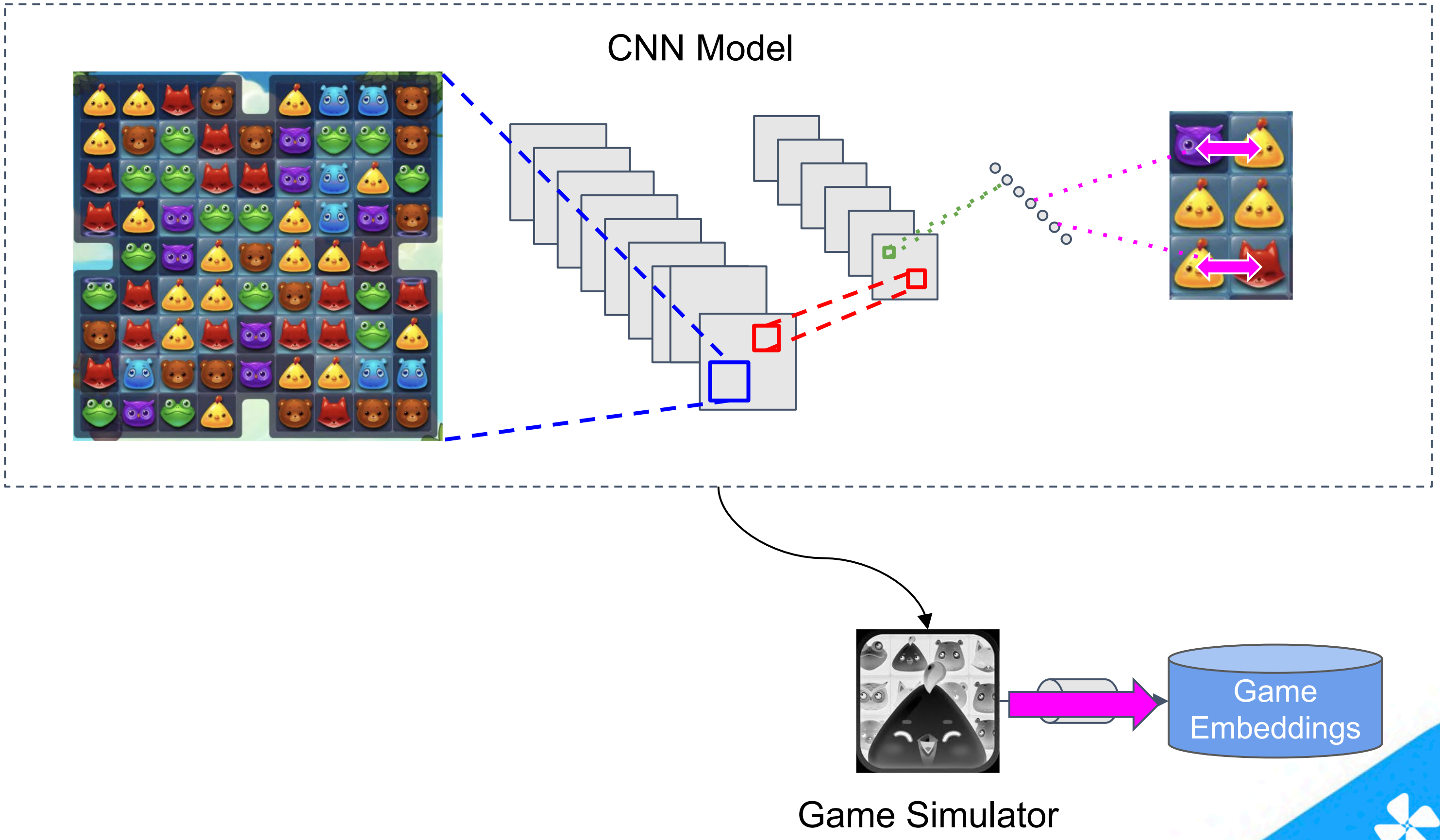
System Architecture: Gameplay Embeddings



System Architecture: Gameplay Embeddings

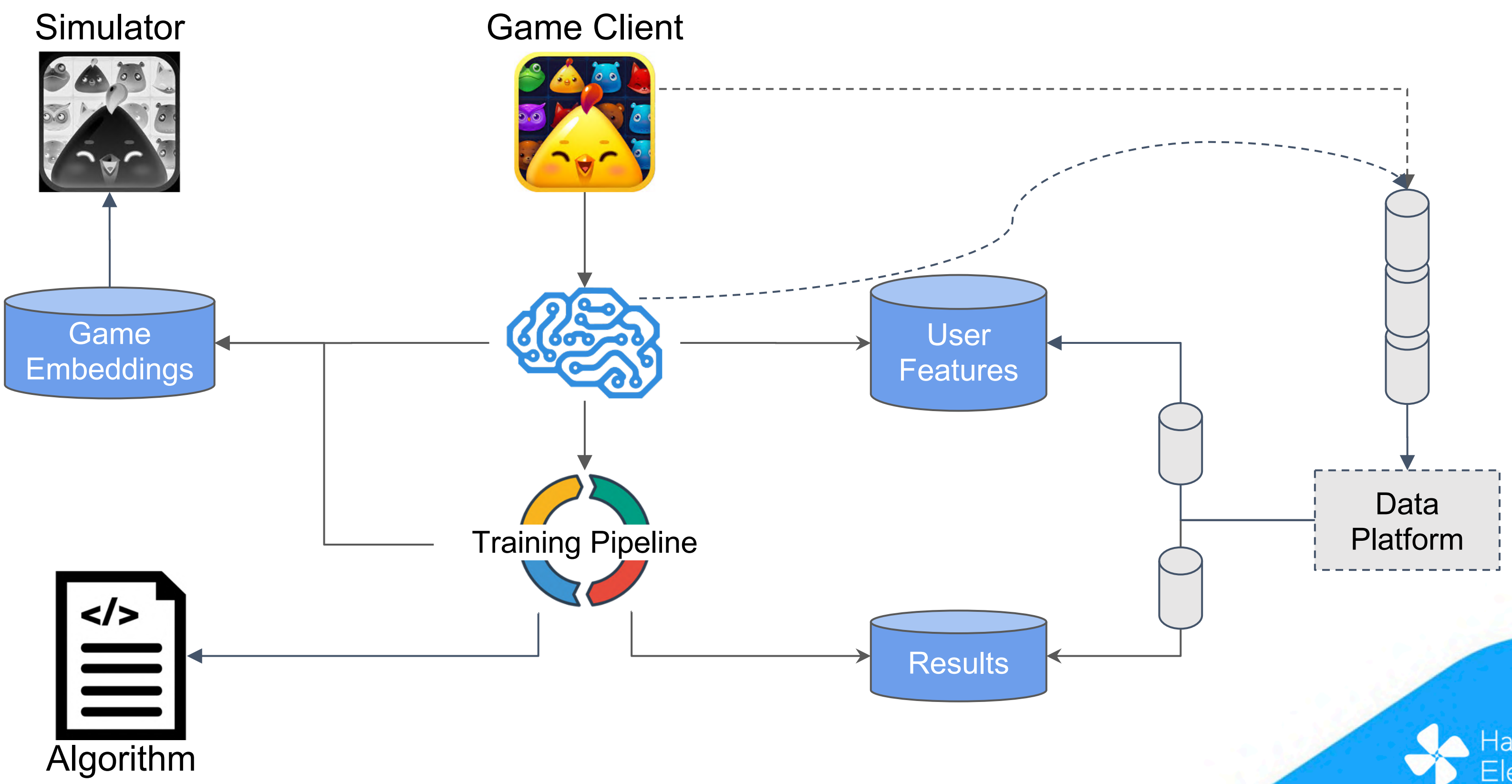


System Architecture: Gameplay Embeddings

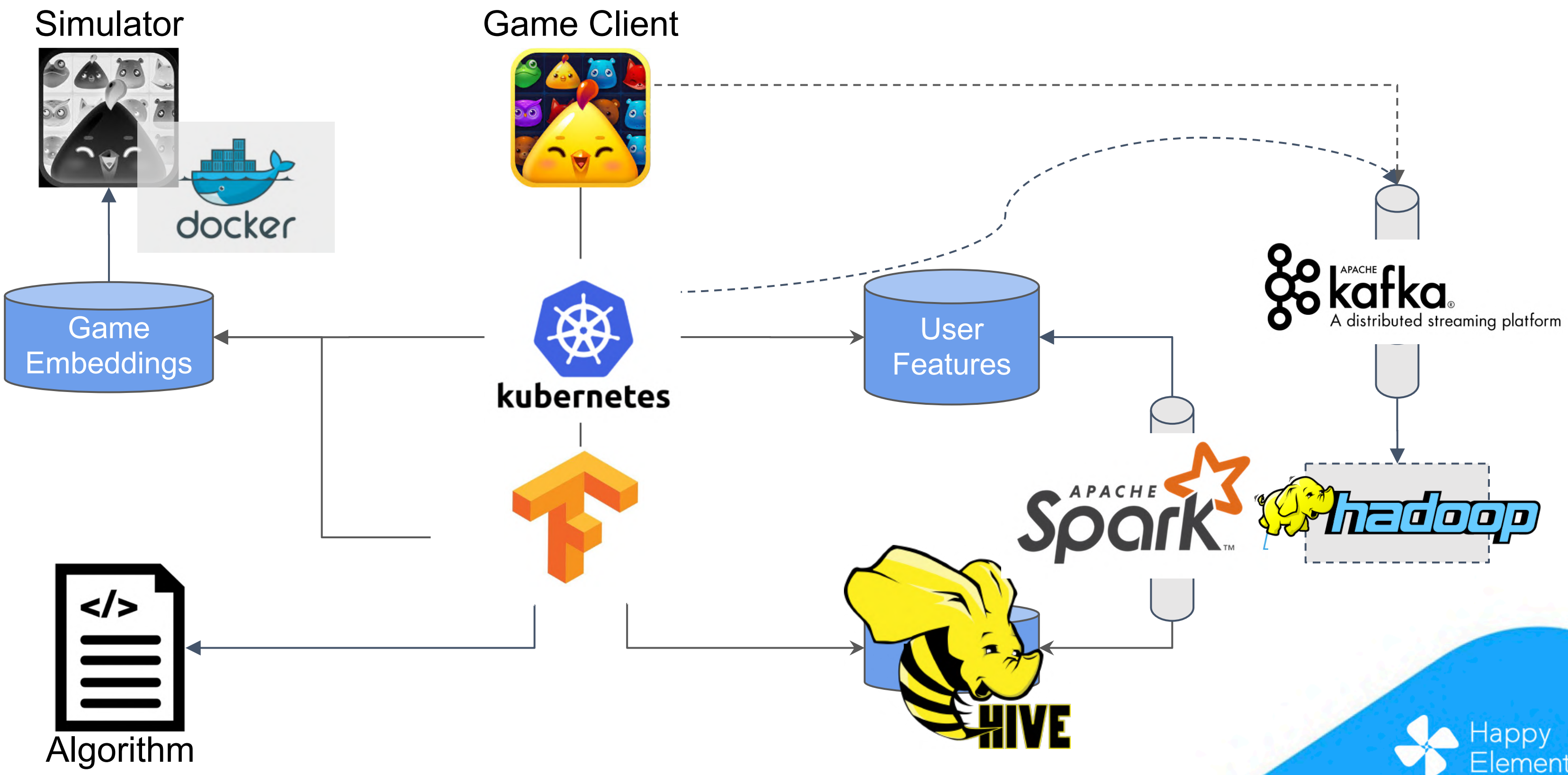


Happy
Elements

System Architecture



Technology Stack Samples



Happy Elements

Conclusions

- Deployment of ML can **significantly improve revenue** and **engagement**
- **Nonstationary** data presents **difficult optimization** problem
- **Relationship** between **short**-term and **long**-term **metrics** hard to identify

Happy Elements

THANKS

http://en.happyelements.com/ai