Test Task Unity Developer [ENG]

Goal

Main card reference

Task:

Use Unity version 2021.2.11f1

Create UI for an "in-hand card" object for CCG-like game. Card consist of:

- Art + UI overlay
- Title
- Description
- Attack icon + text value
- HP icon + text value
- Mana icon + text value

Load card art randomly from https://picsum.photos/ each time app starts.

Fill player's hand with 4-6 cards in a visually pleasing way and use the arc pattern for displaying the cards (look at the pic below). The number of cards should be determined randomly at the start of the game.

Create an UI button at the center of the screen to randomly change one randomly selected value -2→9 (the range is from -2 to 9) of each one card sequentially, starting from the most left card in the player's hand moving right and repeating the sequence after reaching the most right card.

Bind Attack, Health and mana properties to UI. Changing those values from code must be reflected on the card's UI with counter animation. (counting from the initial to the new value)

If some card’s HP drop below 1 - remove this card from player’s hand. (dont forget to reposition other cards, use tweens to make it smooth)

[Middle+] Player can drag a card and drop it on middle section of the table (use drop panel of any size) Card moves back to player’s hand if it’s hasn't been dropped over the drop panel. Cards shines when its being dragged.
Reference: Hearthstone, MTG Arena, Gwent and other same games.
—>

Upload results to GitHub and send the link to hr@gamesredrift.com with ***[Unity Test] <Your Name>*** in title, or directly to the representative.
<—

Treat this task as a real game project scenario. Even if a certain feature or requirement wasn't described clearly and in detail - it doesn't mean you can leave it raw and unfinished.

What's important

1. Project folder & hierarchy structure.
2. Code style, naming, clearness. Proper use of Unity's api. SOLID
3. Logic. Attention to details.
4. UI layout
5. The simplicity and conciseness of the solution is a must.
6. Use of common practices, Optimization.
7. Tween usage.

8. Test completion speed. Overcomplicating solution is a failure.

What's not important

---

1. Visuals used
2. Any non goal related objects or functionality

Good luck!

Use your own art (or unity ui) to perform this task or grab it below:

Red Rift