



**Eco Reward**

**Your project is getting bigger. What should you do with old code, delete it, store it, or improve it?**

**Reward  
Throw Twice**

**Your app runs all day and night, but fewer people use it at night. How could you save energy without shutting it down?**

**Reward**

**Go forward 3 spaces**

**Before sending images to users, should you make them smaller? What are the pros and cons of doing this?**

**Reward  
Double your throw**

**Should your app save all errors or only the most important ones? What are the benefits and risks of each approach?**

**Reward**

**Make another player skip a turn**

**What are the advantages and possible downsides of saving frequently used data for quick access?**

**Reward**

**Go forward 5 spaces**

**For live updates, is it always better to wait for changes instead of checking constantly? When might constant checking be okay?**

**Reward**

**Swap positions with another player**

**How important is it to use eco-friendly data centers? What might make companies choose other options?**

**Reward  
Reduce another  
players next throw  
by 3**



**Should background tasks be done in batches or all the time? When is each method better?**

**Reward  
Throw Twice**

**When choosing tools for your app, should you go for simple, lightweight options or feature-packed ones? What are the trade-offs?**

**Reward**

**Go forward 2 spaces**

**Why is it important to keep an eye on cloud resources? How can you avoid wasting them?**

**Reward**  
**Steal a Eco Reward card from another player**

**Does moving old data  
to storage always save  
money and energy?  
When might it not?**

**Reward  
Double your throw**

**Should you limit the number of times your app asks another service for information? Are there times when frequent requests are needed?**

**Reward**

**Go forward 4 spaces**

**Is adjusting the number of active servers always a better choice than keeping a fixed number? When might a fixed setup be better?**

**Reward  
Throw Twice**

**Should you reduce how often your code gets rebuilt to save resources?**

**How do you decide when a rebuild is actually needed?**

**Reward**

**Go forward 3 spaces**

**When adding a new feature, should you focus more on making it run efficiently or launching it quickly? How do you balance the two?**

**Reward**

**Make another player skip their turn**



**Is using cloud servers  
only when needed  
more energy-efficient  
than keeping them on  
all the time?  
What are the trade-  
offs?**

**Reward  
Steal a Eco Reward  
card from another  
player**

**When managing online servers, what are the benefits and risks of turning off ones that aren't being used?**

**Reward**

**Go forward 5 places**

**Does waiting to load content until it's needed always improve speed and energy use? When might it be a bad idea?**

**Reward**

**Place another player  
4 spaces backwards**

**Should you regularly delete or store old logs?**

**What problems might come from keeping too many or too few?**

**Reward**

**Go forward 2 spaces**

**Is pre-loading web pages a better idea for rarely updated sites? How does it compare to loading pages as needed?**

**Reward  
Throw Twice**

**How does reducing unnecessary small services help with energy use, maintenance, and growth?**

**Reward**

**Place another player  
3 spaces backwards**

**Should automated tests in live apps run less often to save energy, or is frequent testing worth the cost?**

**Reward  
Go forward 3 spaces**

**What are the benefits of putting servers in power-saving mode? Are there times when it's not a good idea?**

**Reward  
Double your throw**



**Is using smaller,  
energy-efficient  
servers always better  
than having a few  
powerful ones?  
How do you balance  
cost, performance, and  
sustainability?**

**Reward  
Go forward 4 spaces**

**Why should  
developers track how  
much energy their  
software uses?  
How can this  
information help them  
make better choices?**

**Reward  
Make another player  
skip a turn**