

Starter Kit

Starter AIs are available in the Starter Kit. They can help you get started with your own bot.

Introductory video by Sebastien and Loïc: https://youtu.be/OJxV_zhICIs

This challenge is inspired from the board game Samsara

Goal

The game takes place in a company IT department. Management is organizing a hackathon on the Green IT theme. Two development teams are competing to be the most efficient team. Release the required applications before the opponent team to have more points, but respect the Green IT requirements or you'll get drowned in the Technical Debt.

Rules

Each player has an IT development team. The game takes place in an office with 8 desks. The hackathon will last many turns. Each turn, the players will play one after the other.

This game is based on **Deck Building**. Each team will have its own deck of cards that will get bigger during the game.

Each team starts the game with **4 BONUS skill** cards and **4 Technical Debt** cards. Those cards will be their personal draw pile. And their personal discard pile will be reshuffled into the draw pile when it gets empty.

The team will get more cards and will lose some during the game.

The game space (the office)

The office is made of 8 desks. Each desk is dedicated to a specific skill

- **TRAINING** (0)
- **CODING** (1)
- **DAILY_ROUTINE** (2)
- **TASK_PRIORITIZATION** (3)
- **ARCHITECTURE_STUDY** (4)
- **CONTINUOUS_INTEGRATION** (5)
- **CODE_REVIEW** (6)
- **REFACTORING** (7)

Those desks are numbered from **0** to **7**. Each of those desks contains **5 skill** cards at the beginning of the game.

Due to covid-19, Management enforced a one way path in the office so as not to bump into other people. You must always move in the same direction.

This forces you to go through the administrative task desk when you move from the desk 7 to the desk 0.

Applications

Each application needs some skills to be released. Applications are the same for both players. Once an application has been released, it is removed from the game for both players.

Applications will get bigger through the leagues.

A small application needs 3 sets of 2 skills (like 2 **REFACTORING**, 2 **TRAINING** and 2 **CODING**)

A big application needs 2 sets of 4 skills (like 4 **DAILY_ROUTINE** and 4 **CODE_REVIEW**)

Each **BONUS skill** card provides one good skill (any of them) and one shoddy skill.

Each specific **skill** card provides 2 good skills (of this specific type) and 2 shoddy skills.

For instance, a **CODING** card provides 2 **CODING** skills and 2 shoddy skills. Those shoddy skills can be used to replace any missing skill, but the quality will not be there.

Each shoddy skill used during an application release will give you a **Technical Debt** card. Those cards are useless. They are only here to slow you down since you can draw some from your draw pile.

Turn Description

At the beginning of your turn, your team draws 4 cards at random from their draw pile.

1. Move

The team begins by moving to another desk (they must leave the desk for the other team).

They will get one **skill** card from the desk they are moving to (TRAINING, CODE_REVIEW, REFACTORING, CODING...).

If the desk is empty (no more cards), the team will get a **BONUS skill** card instead.

If the team arrives on the opponent's desk (or on a desk next to the opponent), they will disturb the opponent (they are noisy when they work).

As an apology, the team must **GIVE** one **skill** card from their hand to the opponent (the team can choose which one).

If they do not have any **skill** card in hand, they get 2 **Technical Debt** cards.

If the team goes through the **administrative task** desk (between the desks 7 and 0), they lose 2 **skills** cards from their hand (they can choose which ones).

If they do not have enough **skill** cards in hand, they get one **Technical Debt** card per missing **skill**.

2. Use a skill

After moving, the team can use one of their available **skills** in hand. This phase will only appear if you can play a skill. The skills and their actions:

- **TRAINING** (0). The team draws 2 cards and can play 1 more card from their hand.
- **CODING** (1). The team draws 1 card and can play 2 more cards from their hand.
- **DAILY_ROUTINE** (2). This is a permanent **skill**: once played, it stays active until the team has released one application.
After moving, the team can get their new **skill** card from a desk one step away. If you play many cards, their effects will add up.
- **TASK_PRIORITIZATION** (3). The team will remove one **skill** card from their hand and put it back on the board. They will then take another available **skill** card from the board.
- **ARCHITECTURE_STUDY** (4). This is a permanent **skill**: once played, it stays active until the team has released one application.
The team will draw one more card at the beginning of their turn. If you play many cards, their effects will add up.
- **CONTINUOUS_INTEGRATION** (5). The team automate one of their **skills** from their hand.
This card will never be discarded and will always be available, but only to release applications.
- **CODE_REVIEW** (6). The team gets 2 **BONUS skill** cards from the board and puts them in their discard pile.
- **REFACTORING** (7). The team can remove a **Technical Debt** card from their hand and put it back on the board.

3. Release an application

The team can use their available **skills** in hand to **RELEASE** an application.

This phase will only appear if you have enough **skills** to do it.

Don't forget, if you use shoddy skills, you'll receive **Technical Debt** cards! When you release an application, your permanent **skills** are discarded.

4. End of the turn

All cards in hand (**skills** and **Technical Debt**) are discarded.

Game end

The hackathon stops as soon as one team releases its 5th application.

Beware, since this hackathon is on the theme of Green IT, the referees will be paying close attention to the quality of the last released application.

The 5th application of each team cannot be released with shoddy skills!

When a team has released 5 applications, the game will end once all teams have played the same amount of turns.

The winner is the team that released the most applications.

In case of a tie, the winning team will be the one that has the less **Technical Debt** cards.

Victory Conditions

- You released 5 applications before your opponent.
- You released more applications than your opponent after **200 game phases**.
- In case of a tie, you have less **Technical Debt** cards than your opponent.

Defeat Conditions

- Your program does not provide a valid command in time.

Debugging tips

- Hover over a player, card or card pile to see extra information about it
- Append text after any command and that text will appear above your player
- Press the gear icon on the viewer to access extra display options
- Use the keyboard to control the action: space to play/pause, arrows to step 1 frame at a time

Technical Details

- The game starts with 5 **skill** cards on each desk, 36 **BONUS** skill cards and 100 **Technical Debt** cards in all the game.
- Each phase is given to the players only if they can perform an action. If they have no choice and the action is mandatory, it will be played automatically.
- If you need to draw a skill card from a desk and there are none left, you will draw a **BONUS** skill card instead. If there is no **BONUS** skill card left, you will get no card.
- In case of a tie, the player who gained the lowest amount of **Technical Debt** will win.
- The 5th application cannot be released with shoddy skills. You will need to have the exact skills.
- The source code of this game is available [on this GitHub repo](#).

Game Input/Output

Input for One Game Turn

Line 1: `gamePhase` the name of the current game phase. It can be `MOVE`, `RELEASE`

Line 2: `applicationCount` for the amount of applications that are still to be released.

Next `applicationCount` lines: the word `APPLICATION` followed by 9 integers, the description of the application to release and the needed skills to release them (`APPLICATION` `applicationId` `trainingNeeded` `codingNeeded` `dailyRoutineNeeded` `taskPrioritizationNeeded` `architectureStudyNeeded` `continuousDeliveryNeeded` `codeReviewNeeded` `refactoringNeeded`).

1 line per player: 4 integers (Your data is always given first):

- `location` : the desk currently used by the player (-1 in the first turn).
- `score` : the amount of released applications.
- `permanentDailyRoutineCards` : the amount of **DAILY_ROUTINE** (2) cards played.
- `permanentArchitectureStudyCards` : the amount of **ARCHITECTURE_STUDY** (2) cards played.

Next line: `cardLocationsCount` for the amount of locations where the players have some cards.

Next `cardLocationsCount` lines: the location name followed by 10 integers, the number of cards in this location (`cardLocation` `trainingCardsCount` `codingCardsCount` `dailyRoutineCardsCount` `taskPrioritizationCardsCount` `architectureStudyCardsCount` `continuousDeliveryCardsCount` `codeReviewCardsCount` `refactoringCardsCount` `bonusCardsCount` `technicalDebtCardsCount`). Location can be `HAND`, `DRAW`, `DISCARD`, `PLAYED_CARDS` the cards you played during this turn and that will go to your discard pile at the end of your turn, `AUTOMATED`, `OPPONENT_CARDS` (all the opponent's cards in their hand, draw and discard) or `OPPONENT_AUTOMATED`

Next line: `possibleMovesCount` for the amount of possible actions.

Next `possibleMovesCount` lines: a string, one possible action.

Output for One Game Turn

1 line containing one of the following actions depending on the game phase:

- `RANDOM`: the player decides to do one of the possible action at random.
- `WAIT`: the player decides not to do any optional action.
- `MOVE` `zoneId`: the player moves to another desk.
This action is mandatory and is only available in the MOVE phase.
- `MOVE` `zoneId` `cardTypeToTake`: the player moves to another desk and gets a skill card from another desk.
This action is mandatory and is only available in the MOVE phase and if the player has already played some DAILY_ROUTINE cards.
- `GIVE` `cardTypeId`: the player gives a skill card from their hand to their opponent.
This action is mandatory and is only available in the GIVE_CARD phase.
- `THROW` `cardTypeId`: the player puts back a skill card from their hand to the game board.
This action is mandatory and is only available in the THROW_CARD phase.
- `TRAINING`: the player plays a TRAINING card from their hand.
This action is optional and is only available in the PLAY_CARD phase.
- `CODING`: the player plays a CODING card from their hand.
This action is optional and is only available in the PLAY_CARD phase.
- `DAILY_ROUTINE`: the player plays a DAILY_ROUTINE card from their hand.
This action is optional and is only available in the PLAY_CARD phase.
- `TASK_PRIORITIZATION` `cardTypeToThrow` `cardTypeToTake`: the player plays a TASK_PRIORITIZATION card from their hand.
This action is optional and is only available in the PLAY_CARD phase.

- **ARCHITECTURE_STUDY**: the player plays a ARCHITECTURE_STUDY card from their hand. This action is optional and is only available in the PLAY_CARD phase.
- **CONTINUOUS_INTEGRATION** **cardTypeToAutomate**: the player plays a CONTINUOUS_INTEGRATION card from their hand. This action is optional and is only available in the PLAY_CARD phase.
- **CODE_REVIEW**: the player plays a CODE_REVIEW card from their hand. This action is optional and is only available in the PLAY_CARD phase.
- **REFACTORING**: the player plays a REFACTORING card from their hand. This action is optional and is only available in the PLAY_CARD phase.
- **RELEASE** **applicationId**: the player releases an application. This action is optional and is only available in the RELEASE phase.

You may append text to a command to have it displayed in the viewer above your player.

Examples:

- **MOVE 3**
- **RELEASE 16**
- **GIVE 8 a gift for you!**
- **REFACTORING goodbye technical debt!**
- **CONTINUOUS_INTEGRATION 1**
- **TASK_PRIORITIZATION 8 2**
- **WAIT nothing to do...**
- **RANDOM got no idea...**

Constraints

Response time per turn ≤ 50 ms (Doing Green IT means sparing resources)
 Response time for the first turn ≤ 1000 ms

If you want to know more on Green IT (in French)

Societe Generale ambitions on Green IT : <https://youtu.be/ZstnO7j1y4c>

Our partnership with INR and our signing of the Sustainable IT Charter : <https://careers.societegenerale.com/green-it-program>

Our experts Masterclasses in video – to know everything on Green IT

Masterclass #1 Why doing Sustainable IT? : <https://youtu.be/eLffG8Z0iXU>

Masterclass #2 E-accessibility stakes for IT : https://youtu.be/oRA_CrGxGgw

Masterclass #3 Architecture and Green IT : <https://youtu.be/x2fMjGqinLA>

Masterclass #4 Moving to a sustainable IT design : <https://youtu.be/kb2PM7OniRk>