[](https://ida.interchain.io/)

[Interchain Developer Academy](https://ida.interchain.io/)/[Interchain Developer Academy](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/4-game-forfeit.html)



Search

[Interchain Developer Academy](https://ida.interchain.io/)[Interchain Developer Academy](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/4-game-forfeit.html)

Search



Filters

Interchain Developer Academy

[](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/4-game-forfeit.html)

[Week 0 - Getting Started](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/4-game-forfeit.html)

[Getting Started](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/4-game-forfeit.html)

[Blockchain 101](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/4-game-forfeit.html)

[Blockchain History](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/4-game-forfeit.html)

[Public and Managed Blockchains](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/4-game-forfeit.html)

[Consensus in Distributed Networks](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/4-game-forfeit.html)

[Cryptography](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/4-game-forfeit.html)

[Self-Assessment Quiz](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/4-game-forfeit.html)

[Go Introduction - First Steps](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/4-game-forfeit.html)

[Go Basics](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/4-game-forfeit.html)

[Go Interfaces](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/4-game-forfeit.html)

[Control Structures in Go](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/4-game-forfeit.html)

[Arrays and Slices in Go](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/4-game-forfeit.html)

[Standard Packages in Go](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/4-game-forfeit.html)

[Concurrency in Go](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/4-game-forfeit.html)

[Good-To-Know Dev Terms](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/4-game-forfeit.html)

[Docker Introduction](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/4-game-forfeit.html)

[](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/4-game-forfeit.html)

[Week 1 - Introduction to the Interchain](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/4-game-forfeit.html)

[Introduction to the Interchain](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/4-game-forfeit.html)

[Blockchain Technology and the Interchain](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/4-game-forfeit.html)

[The Interchain Ecosystem](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/4-game-forfeit.html)

[Getting ATOM and Staking It](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/4-game-forfeit.html)

[A Blockchain App Architecture](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/4-game-forfeit.html)

[Accounts](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/4-game-forfeit.html)

[Transactions](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/4-game-forfeit.html)

[Messages](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/4-game-forfeit.html)

[Modules](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/4-game-forfeit.html)

[Protobuf](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/4-game-forfeit.html)

[Multistore and Keepers](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/4-game-forfeit.html)

[BaseApp](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/4-game-forfeit.html)

[Queries](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/4-game-forfeit.html)

[Events](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/4-game-forfeit.html)

[Context](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/4-game-forfeit.html)

[Testing](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/4-game-forfeit.html)

[Relaying with IBC](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/4-game-forfeit.html)

[Interchain Security](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/4-game-forfeit.html)

[Bridges](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/4-game-forfeit.html)

[Migrations](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/4-game-forfeit.html)

[Week 1 Quiz](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/4-game-forfeit.html)

[](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/4-game-forfeit.html)

[Week 2 - First Steps](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/4-game-forfeit.html)

[First Steps](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/4-game-forfeit.html)

[Setup Your Work Environment](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/4-game-forfeit.html)

[Run a Node, API, and CLI](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/4-game-forfeit.html)

[Ignite CLI](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/4-game-forfeit.html)

[Exercise - Make a Checkers Blockchain](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/4-game-forfeit.html)

[Store Object](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/4-game-forfeit.html)

[Create Custom Messages](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/4-game-forfeit.html)

[Create and Save a Game Properly](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/4-game-forfeit.html)

[Add a Way to Make a Move](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/4-game-forfeit.html)

[Emit Game Information](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/4-game-forfeit.html)

[Record the Game Winner](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/4-game-forfeit.html)

[Week 2 Exercise](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/4-game-forfeit.html)

[](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/4-game-forfeit.html)

[Week 3 - Introduction to IBC and CosmJS](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/4-game-forfeit.html)

[Introduction to IBC and CosmJS](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/4-game-forfeit.html)

[What is IBC?](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/4-game-forfeit.html)

[IBC/TAO - Connections (OPTIONAL)](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/4-game-forfeit.html)

[IBC/TAO - Channels (OPTIONAL)](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/4-game-forfeit.html)

[IBC/TAO - Clients (OPTIONAL)](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/4-game-forfeit.html)

[IBC Token Transfer](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/4-game-forfeit.html)

[Interchain Accounts (OPTIONAL)](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/4-game-forfeit.html)

[IBC Middleware (OPTIONAL)](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/4-game-forfeit.html)

[Create a Custom IBC Middleware (OPTIONAL)](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/4-game-forfeit.html)

[Integrate IBC Middleware Into a Chain (OPTIONAL)](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/4-game-forfeit.html)

[IBC Tooling](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/4-game-forfeit.html)

[What is CosmJS?](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/4-game-forfeit.html)

[Your First CosmJS Actions](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/4-game-forfeit.html)

[Compose Complex Transactions](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/4-game-forfeit.html)

[Learn to Integrate Keplr](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/4-game-forfeit.html)

[Create Custom CosmJS Interfaces](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/4-game-forfeit.html)

[](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/4-game-forfeit.html)

[Week 4 - Ignite CLI and IBC Advanced](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/4-game-forfeit.html)

[Ignite CLI and IBC Advanced](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/4-game-forfeit.html)

[Keep an Up-To-Date Game Deadline](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/4-game-forfeit.html)

[Keep Track Of How Many Moves Have Been Played](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/4-game-forfeit.html)

[Put Your Games in Order](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/4-game-forfeit.html)

[Auto-Expiring Games](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/4-game-forfeit.html)

[Let Players Set a Wager](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/4-game-forfeit.html)

[Handle wager payments](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/4-game-forfeit.html)

[Integration tests](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/4-game-forfeit.html)

[Incentivize Players](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/4-game-forfeit.html)

[Help Find a Correct Move](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/4-game-forfeit.html)

[Play With Cross-Chain Tokens](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/4-game-forfeit.html)

[Understand IBC Denoms](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/4-game-forfeit.html)

[Go Relayer](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/4-game-forfeit.html)

[Hermes Relayer](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/4-game-forfeit.html)

[](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/4-game-forfeit.html)

[Week 5 - CosmJS Advanced](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/4-game-forfeit.html)

[CosmJS Advanced](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/4-game-forfeit.html)

[Create Custom Objects](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/4-game-forfeit.html)

[Create Custom Messages](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/4-game-forfeit.html)

[Get an External GUI](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/4-game-forfeit.html)

[Integrate CosmJS and Keplr](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/4-game-forfeit.html)

[Backend Script for Game Indexing](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/4-game-forfeit.html)

[](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/4-game-forfeit.html)

[Week 6 - IBC Deep Dive](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/4-game-forfeit.html)

[IBC Deep Dive](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/4-game-forfeit.html)

[IBC Application Developer Introduction](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/4-game-forfeit.html)

[Make a Module IBC-Enabled](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/4-game-forfeit.html)

[Adding Packet and Acknowledgment Data](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/4-game-forfeit.html)

[Extend the Checkers Game With a Leaderboard](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/4-game-forfeit.html)

[Create a Leaderboard Chain](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/4-game-forfeit.html)

[](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/4-game-forfeit.html)

[Week 7 - From Code to MVP to Production and Migrations](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/4-game-forfeit.html)

[From Code to MVP to Production and Migrations](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/4-game-forfeit.html)

[Run in Production](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/4-game-forfeit.html)

[Prepare the Software to Run](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/4-game-forfeit.html)

[Prepare a Validator and Keys](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/4-game-forfeit.html)

[Prepare Where the Node Starts](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/4-game-forfeit.html)

[Prepare and Connect to Other Nodes](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/4-game-forfeit.html)

[Configure, Run, and Set Up a Service](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/4-game-forfeit.html)

[Prepare and Do Migrations](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/4-game-forfeit.html)

[Simulate Production in Docker](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/4-game-forfeit.html)

[Tally Player Info After Production](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/4-game-forfeit.html)

[Add a Leaderboard as a Module](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/4-game-forfeit.html)

[Migrate the Leaderboard Module After Production](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/4-game-forfeit.html)

[Simulate a Migration in Docker](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/4-game-forfeit.html)

[Final Exam](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/4-game-forfeit.html)

[](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/4-game-forfeit.html)

[What's Next?](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/4-game-forfeit.html)

[Continue Your Interchain Journey](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/4-game-forfeit.html)

Docs Version Switcher

On this page

[New information](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/4-game-forfeit.html#new-information)

[Putting callbacks in place](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/4-game-forfeit.html#putting-callbacks-in-place)

[Expire games handler](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/4-game-forfeit.html#expire-games-handler)

[Prepare the main loop](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/4-game-forfeit.html#prepare-the-main-loop)

[Identify an expired game](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/4-game-forfeit.html#identify-an-expired-game)

[Handle an expired game](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/4-game-forfeit.html#handle-an-expired-game)

[Unit tests](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/4-game-forfeit.html#unit-tests)

[Interact via the CLI](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/4-game-forfeit.html#interact-via-the-cli)

[#Copy link](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/4-game-forfeit.html#auto-expiring-games) **Auto-Expiring Games**



Make sure you have everything you need before proceeding:

* You understand the concepts of [ABCI](https://ida.interchain.io/academy/2-cosmos-concepts/1-architecture.html).
* Go is installed.
* You have the checkers blockchain codebase with the elements necessary for forfeit. If not, follow the [previous steps](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/3-game-fifo.html) or check out [the relevant version (opens new window)↗](https://github.com/cosmos/b9-checkers-academy-draft/tree/game-fifo).



In this section, you will:

* Do begin block and end block operations.
* Forfeit games automatically.
* Do garbage collection.

In the [previous section](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/3-game-fifo.html) you prepared the expiration of games:

* A First-In-First-Out (FIFO) that always has old games at its head and freshly updated games at its tail.
* A deadline field to guide the expiration.
* A winner field to further assist with forfeiting.
* A move count field to inform the action to take when forfeiting.

[#Copy link](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/4-game-forfeit.html#new-information) New information

A game expires in two different situations:

1. It was never really played, so it is removed quietly. That includes a single move by a single player.
2. Moves were played by both players, making it a proper game, and forfeit is the outcome because a player then failed to play a move in time.

In the latter case, you want to emit a new event which differentiates forfeiting a game from a win involving a move. Therefore you define new error constants:



Copy

const (

GameForfeitedEventType = "game-forfeited"

GameForfeitedEventGameIndex = "game-index"

GameForfeitedEventWinner = "winner"

GameForfeitedEventBoard = "board"

)

x /

checkers /

types /

keys.go

[View source→](https://github.com/cosmos/b9-checkers-academy-draft/blob/forfeit-game/x/checkers/types/keys.go" \l "L57-L62" \t "_blank)

[#Copy link](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/4-game-forfeit.html#putting-callbacks-in-place) Putting callbacks in place

When you use Ignite CLI to scaffold your module, it creates the [x/checkers/module.go (opens new window)↗](https://github.com/cosmos/b9-checkers-academy-draft/blob/forfeit-game/x/checkers/module.go) file with a lot of functions to accommodate your application. In particular, the function that **may** be called on your module on EndBlock is named EndBlock:



Copy

func (am AppModule) EndBlock(\_ sdk.Context, \_ abci.RequestEndBlock) []abci.ValidatorUpdate {

return []abci.ValidatorUpdate{}

}

x /

checkers /

module.go

[View source→](https://github.com/cosmos/b9-checkers-academy-draft/blob/game-fifo/x/checkers/module.go" \l "L173-L175" \t "_blank)

Ignite CLI left this empty. It is here that you add what you need done right before the block gets sealed. Create a new file named x/checkers/keeper/end\_block\_server\_game.go to encapsulate the knowledge about game expiry. Leave your function empty for now:



Copy

func (k Keeper) ForfeitExpiredGames(goCtx context.Context) {

// TODO

}

x /

checkers /

keeper /

end\_block\_server\_game.go

[View source→](https://github.com/cosmos/b9-checkers-academy-draft/blob/forfeit-game/x/checkers/keeper/end_block_server_game.go" \l "L12" \t "_blank)

In x/checkers/module.go update EndBlock with:



Copy

func (am AppModule) EndBlock(ctx sdk.Context, \_ abci.RequestEndBlock) []abci.ValidatorUpdate {

+ am.keeper.ForfeitExpiredGames(sdk.WrapSDKContext(ctx))

return []abci.ValidatorUpdate{}

}

x /

checkers /

module.go

[View source→](https://github.com/cosmos/b9-checkers-academy-draft/blob/forfeit-game/x/checkers/module.go" \l "L174" \t "_blank)

This ensures that **if** your module's EndBlock function is called the expired games will be handled. For the **whole application to call your module** you have to instruct it to do so. This takes place in app/app.go, where the application is initialized with the proper order to call the EndBlock functions in different modules. In fact, yours has already been placed at the end by Ignite:



Copy

app.mm.SetOrderEndBlockers(

crisistypes.ModuleName,

...

+ checkersmoduletypes.ModuleName,

)

app /

app.go

[View source→](https://github.com/cosmos/b9-checkers-academy-draft/blob/forfeit-game/app/app.go" \l "L493" \t "_blank)

Your ForfeitExpiredGames function will now be called at the end of each block.

Also prepare a new error:



Copy

var (

...

+ ErrCannotFindWinnerByColor = sdkerrors.Register(ModuleName, 1109, "cannot find winner by color: %s")

)

x /

checkers /

types /

errors.go

[View source→](https://github.com/cosmos/b9-checkers-academy-draft/blob/forfeit-game/x/checkers/types/errors.go" \l "L23" \t "_blank)

[#Copy link](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/4-game-forfeit.html#expire-games-handler) Expire games handler

With the callbacks in place, it is time to code the expiration properly.

[#Copy link](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/4-game-forfeit.html#prepare-the-main-loop) Prepare the main loop

In ForfeitExpiredGames, it is a matter of looping through the FIFO, starting from the head, and handling games that are expired. You can stop at the first active game, as all those that come after are also active thanks to the careful updating of the FIFO.

1. Prepare useful information:



Copy

ctx := sdk.UnwrapSDKContext(goCtx)

opponents := map[string]string{

rules.PieceStrings[rules.BLACK\_PLAYER]: rules.PieceStrings[rules.RED\_PLAYER],

rules.PieceStrings[rules.RED\_PLAYER]: rules.PieceStrings[rules.BLACK\_PLAYER],

}

x /

checkers /

keeper /

end\_block\_server\_game.go

[View source→](https://github.com/cosmos/b9-checkers-academy-draft/blob/forfeit-game/x/checkers/keeper/end_block_server_game.go" \l "L13-L18" \t "_blank)

1. Initialize the parameters before entering the loop:



Copy

systemInfo, found := k.GetSystemInfo(ctx)

if !found {

panic("SystemInfo not found")

}

gameIndex := systemInfo.FifoHeadIndex

var storedGame types.StoredGame

x /

checkers /

keeper /

end\_block\_server\_game.go

[View source→](https://github.com/cosmos/b9-checkers-academy-draft/blob/forfeit-game/x/checkers/keeper/end_block_server_game.go" \l "L21-L27" \t "_blank)

1. Enter the loop:



Copy

for {

// TODO

}

x /

checkers /

keeper /

end\_block\_server\_game.go

[View source→](https://github.com/cosmos/b9-checkers-academy-draft/blob/forfeit-game/x/checkers/keeper/end_block_server_game.go" \l "L28" \t "_blank)

See below for what replaces this TODO.

1. After the loop has ended do not forget to save the latest FIFO state:



Copy

k.SetSystemInfo(ctx, systemInfo)

x /

checkers /

keeper /

end\_block\_server\_game.go

[View source→](https://github.com/cosmos/b9-checkers-academy-draft/blob/forfeit-game/x/checkers/keeper/end_block_server_game.go" \l "L71" \t "_blank)

So what goes in the for { TODO }?

[#Copy link](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/4-game-forfeit.html#identify-an-expired-game) Identify an expired game

1. Start with a loop breaking condition, if your cursor has reached the end of the FIFO:



Copy

if gameIndex == types.NoFifoIndex {

break

}

x /

checkers /

keeper /

end\_block\_server\_game.go

[View source→](https://github.com/cosmos/b9-checkers-academy-draft/blob/forfeit-game/x/checkers/keeper/end_block_server_game.go" \l "L30-L32" \t "_blank)

1. Fetch the expired game candidate and its deadline:



Copy

storedGame, found = k.GetStoredGame(ctx, gameIndex)

if !found {

panic("Fifo head game not found " + systemInfo.FifoHeadIndex)

}

deadline, err := storedGame.GetDeadlineAsTime()

if err != nil {

panic(err)

}

x /

checkers /

keeper /

end\_block\_server\_game.go

[View source→](https://github.com/cosmos/b9-checkers-academy-draft/blob/forfeit-game/x/checkers/keeper/end_block_server_game.go" \l "L33-L40" \t "_blank)

1. Test for expiration:



Copy

if deadline.Before(ctx.BlockTime()) {

// TODO

} else {

// All other games after are active anyway

break

}

x /

checkers /

keeper /

end\_block\_server\_game.go

[View source→](https://github.com/cosmos/b9-checkers-academy-draft/blob/forfeit-game/x/checkers/keeper/end_block_server_game.go" \l "L41" \t "_blank)

Now, what goes into this if "expired" { TODO }?

[#Copy link](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/4-game-forfeit.html#handle-an-expired-game) Handle an expired game

1. If the game has expired, remove it from the FIFO:



Copy

k.RemoveFromFifo(ctx, &storedGame, &systemInfo)

x /

checkers /

keeper /

end\_block\_server\_game.go

[View source→](https://github.com/cosmos/b9-checkers-academy-draft/blob/forfeit-game/x/checkers/keeper/end_block_server_game.go" \l "L43" \t "_blank)

1. Check whether the game is worth keeping. If it is, set the winner as the opponent of the player whose turn it is, remove the board, and save:



Copy

lastBoard := storedGame.Board

if storedGame.MoveCount <= 1 {

// No point in keeping a game that was never really played

k.RemoveStoredGame(ctx, gameIndex)

} else {

storedGame.Winner, found = opponents[storedGame.Turn]

if !found {

panic(fmt.Sprintf(types.ErrCannotFindWinnerByColor.Error(), storedGame.Turn))

}

storedGame.Board = ""

k.SetStoredGame(ctx, storedGame)

}

x /

checkers /

keeper /

end\_block\_server\_game.go

[View source→](https://github.com/cosmos/b9-checkers-academy-draft/blob/forfeit-game/x/checkers/keeper/end_block_server_game.go" \l "L44-L55" \t "_blank)

1. Emit the relevant event:



Copy

ctx.EventManager().EmitEvent(

sdk.NewEvent(types.GameForfeitedEventType,

sdk.NewAttribute(types.GameForfeitedEventGameIndex, gameIndex),

sdk.NewAttribute(types.GameForfeitedEventWinner, storedGame.Winner),

sdk.NewAttribute(types.GameForfeitedEventBoard, lastBoard),

),

)

x /

checkers /

keeper /

end\_block\_server\_game.go

[View source→](https://github.com/cosmos/b9-checkers-academy-draft/blob/forfeit-game/x/checkers/keeper/end_block_server_game.go" \l "L56-L62" \t "_blank)

1. Move along the FIFO for the next run of the loop:



Copy

gameIndex = systemInfo.FifoHeadIndex

x /

checkers /

keeper /

end\_block\_server\_game.go

[View source→](https://github.com/cosmos/b9-checkers-academy-draft/blob/forfeit-game/x/checkers/keeper/end_block_server_game.go" \l "L64" \t "_blank)



For an explanation as to why this setup is resistant to an attack from an unbounded number of expired games, see the [section on the game's FIFO](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/3-game-fifo.html).

[#Copy link](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/4-game-forfeit.html#unit-tests) Unit tests

How do you test something that is supposed to happen during the EndBlock event? You call the function that will be called within EndBlock (i.e. Keeper.ForfeitExpiredGames). Create a new test file end\_block\_server\_game\_test.go for your tests. The situations that you can test are:

1. A game was never played, while alone in the state [or not (opens new window)↗](https://github.com/cosmos/b9-checkers-academy-draft/blob/forfeit-game/x/checkers/keeper/end_block_server_game_test.go#L44-L79). Or [two games (opens new window)↗](https://github.com/cosmos/b9-checkers-academy-draft/blob/forfeit-game/x/checkers/keeper/end_block_server_game_test.go#L81-L130) were never played. In this case, you need to confirm that the game was fully deleted, and that an event was emitted with no winners:



Copy

func TestForfeitUnplayed(t \*testing.T) {

\_, keeper, context := setupMsgServerWithOneGameForPlayMove(t)

ctx := sdk.UnwrapSDKContext(context)

game1, found := keeper.GetStoredGame(ctx, "1")

require.True(t, found)

game1.Deadline = types.FormatDeadline(ctx.BlockTime().Add(time.Duration(-1)))

keeper.SetStoredGame(ctx, game1)

keeper.ForfeitExpiredGames(context)

\_, found = keeper.GetStoredGame(ctx, "1")

require.False(t, found)

systemInfo, found := keeper.GetSystemInfo(ctx)

require.True(t, found)

require.EqualValues(t, types.SystemInfo{

NextId: 2,

FifoHeadIndex: "-1",

FifoTailIndex: "-1",

}, systemInfo)

events := sdk.StringifyEvents(ctx.EventManager().ABCIEvents())

require.Len(t, events, 2)

event := events[0]

require.EqualValues(t, sdk.StringEvent{

Type: "game-forfeited",

Attributes: []sdk.Attribute{

{Key: "game-index", Value: "1"},

{Key: "winner", Value: "\*"},

{Key: "board", Value: "\*b\*b\*b\*b|b\*b\*b\*b\*|\*b\*b\*b\*b|\*\*\*\*\*\*\*\*|\*\*\*\*\*\*\*\*|r\*r\*r\*r\*|\*r\*r\*r\*r|r\*r\*r\*r\*"},

},

}, event)

}

x /

checkers /

keeper /

end\_block\_server\_game\_test.go

[View source→](https://github.com/cosmos/b9-checkers-academy-draft/blob/forfeit-game/x/checkers/keeper/end_block_server_game_test.go" \l "L12-L42" \t "_blank)

1. A game was played with only one move, while alone in the state [or not (opens new window)↗](https://github.com/cosmos/b9-checkers-academy-draft/blob/forfeit-game/x/checkers/keeper/end_block_server_game_test.go#L172-L215). Or [two games (opens new window)↗](https://github.com/cosmos/b9-checkers-academy-draft/blob/forfeit-game/x/checkers/keeper/end_block_server_game_test.go#L217-L283) were played in this way. In this case, you need to confirm that the game was fully deleted, and that an event was emitted with no winners:



Copy

func TestForfeitPlayedOnce(t \*testing.T) {

msgServer, keeper, context := setupMsgServerWithOneGameForPlayMove(t)

ctx := sdk.UnwrapSDKContext(context)

msgServer.PlayMove(context, &types.MsgPlayMove{

Creator: bob,

GameIndex: "1",

FromX: 1,

FromY: 2,

ToX: 2,

ToY: 3,

})

game1, found := keeper.GetStoredGame(ctx, "1")

require.True(t, found)

game1.Deadline = types.FormatDeadline(ctx.BlockTime().Add(time.Duration(-1)))

keeper.SetStoredGame(ctx, game1)

keeper.ForfeitExpiredGames(context)

\_, found = keeper.GetStoredGame(ctx, "1")

require.False(t, found)

systemInfo, found := keeper.GetSystemInfo(ctx)

require.True(t, found)

require.EqualValues(t, types.SystemInfo{

NextId: 2,

FifoHeadIndex: "-1",

FifoTailIndex: "-1",

}, systemInfo)

events := sdk.StringifyEvents(ctx.EventManager().ABCIEvents())

require.Len(t, events, 3)

event := events[0]

require.EqualValues(t, sdk.StringEvent{

Type: "game-forfeited",

Attributes: []sdk.Attribute{

{Key: "game-index", Value: "1"},

{Key: "winner", Value: "\*"},

{Key: "board", Value: "\*b\*b\*b\*b|b\*b\*b\*b\*|\*\*\*b\*b\*b|\*\*b\*\*\*\*\*|\*\*\*\*\*\*\*\*|r\*r\*r\*r\*|\*r\*r\*r\*r|r\*r\*r\*r\*"},

},

}, event)

}

x /

checkers /

keeper /

end\_block\_server\_game\_test.go

[View source→](https://github.com/cosmos/b9-checkers-academy-draft/blob/forfeit-game/x/checkers/keeper/end_block_server_game_test.go" \l "L132-L170" \t "_blank)

1. A game was played with at least two moves, while alone in the state [or not (opens new window)↗](https://github.com/cosmos/b9-checkers-academy-draft/blob/forfeit-game/x/checkers/keeper/end_block_server_game_test.go#L346-L410). Or [two games (opens new window)↗](https://github.com/cosmos/b9-checkers-academy-draft/blob/forfeit-game/x/checkers/keeper/end_block_server_game_test.go#L412-L520) were played in this way. In this case, you need to confirm the game was not deleted, and instead that a winner was announced, including in events:



Copy

func TestForfeitPlayedTwice(t \*testing.T) {

msgServer, keeper, context := setupMsgServerWithOneGameForPlayMove(t)

ctx := sdk.UnwrapSDKContext(context)

msgServer.PlayMove(context, &types.MsgPlayMove{

Creator: bob,

GameIndex: "1",

FromX: 1,

FromY: 2,

ToX: 2,

ToY: 3,

})

msgServer.PlayMove(context, &types.MsgPlayMove{

Creator: carol,

GameIndex: "1",

FromX: 0,

FromY: 5,

ToX: 1,

ToY: 4,

})

game1, found := keeper.GetStoredGame(ctx, "1")

require.True(t, found)

oldDeadline := types.FormatDeadline(ctx.BlockTime().Add(time.Duration(-1)))

game1.Deadline = oldDeadline

keeper.SetStoredGame(ctx, game1)

keeper.ForfeitExpiredGames(context)

game1, found = keeper.GetStoredGame(ctx, "1")

require.True(t, found)

require.EqualValues(t, types.StoredGame{

Index: "1",

Board: "",

Turn: "b",

Black: bob,

Red: carol,

Winner: "r",

Deadline: oldDeadline,

MoveCount: uint64(2),

BeforeIndex: "-1",

AfterIndex: "-1",

}, game1)

systemInfo, found := keeper.GetSystemInfo(ctx)

require.True(t, found)

require.EqualValues(t, types.SystemInfo{

NextId: 2,

FifoHeadIndex: "-1",

FifoTailIndex: "-1",

}, systemInfo)

events := sdk.StringifyEvents(ctx.EventManager().ABCIEvents())

require.Len(t, events, 3)

event := events[0]

require.EqualValues(t, sdk.StringEvent{

Type: "game-forfeited",

Attributes: []sdk.Attribute{

{Key: "game-index", Value: "1"},

{Key: "winner", Value: "r"},

{Key: "board", Value: "\*b\*b\*b\*b|b\*b\*b\*b\*|\*\*\*b\*b\*b|\*\*b\*\*\*\*\*|\*r\*\*\*\*\*\*|\*\*r\*r\*r\*|\*r\*r\*r\*r|r\*r\*r\*r\*"},

},

}, event)

}

Expand



x /

checkers /

keeper /

end\_block\_server\_game\_test.go

[View source→](https://github.com/cosmos/b9-checkers-academy-draft/blob/forfeit-game/x/checkers/keeper/end_block_server_game_test.go" \l "L285-L344" \t "_blank)



Note how all the attributes of an event of a given type (such as "game-forfeited") aggregate in a single array. The context is not reset on a new transaction, so when testing attributes you either have to compare the full array or take slices to compare what matters.

[#Copy link](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/4-game-forfeit.html#interact-via-the-cli) Interact via the CLI

Currently, the game expiry is one day in the future. This is too long to test with the CLI. Temporarily set it to 5 minutes:



Copy

MaxTurnDuration = time.Duration(5 \* 60 \* 1000\_000\_000) // 5 minutes

x /

checkers /

types /

keys.go

[View source→](https://github.com/cosmos/b9-checkers-academy-draft/blob/forfeit-game/x/checkers/types/keys.go" \l "L49" \t "_blank)

Avoid having games in the FIFO that expire in a day because of your earlier tests:

**Local**

**Docker**



Copy

$ ignite chain serve --reset-once

Copy

$ docker run --rm -it \

--name checkers \

-v $(pwd):/checkers \

-w /checkers \

checkers\_i \

ignite chain serve --reset-once

Export your aliases again:

**Local**

**Docker**



Copy

$ export alice=$(checkersd keys show alice -a)

$ export bob=$(checkersd keys show bob -a)

Copy

$ export alice=$(docker exec checkers checkersd keys show alice -a)

$ export bob=$(docker exec checkers checkersd keys show bob -a)

Create three games one minute apart. Have Alice play the middle one, and both Alice and Bob play the last one:

1

First game:

**Local**

**Docker**



Copy

$ checkersd tx checkers create-game $alice $bob --from $alice

Copy

$ docker exec -it checkers \

checkersd tx checkers create-game $alice $bob --from $alice

2

Wait a minute, then create your second game and play it:

**Local**

**Docker**



Copy

$ checkersd tx checkers create-game $alice $bob --from $bob

$ checkersd tx checkers play-move 2 1 2 2 3 --from $alice

Copy

$ docker exec -it checkers \

checkersd tx checkers create-game $alice $bob --from $bob

$ docker exec -it checkers \

checkersd tx checkers play-move 2 1 2 2 3 --from $alice

3

Wait another minute, then create your third game and play on it:

**Local**

**Docker**



Copy

$ checkersd tx checkers create-game $alice $bob --from $alice

$ checkersd tx checkers play-move 3 1 2 2 3 --from $alice

$ checkersd tx checkers play-move 3 0 5 1 4 --from $bob

Copy

$ docker exec -it checkers \

checkersd tx checkers create-game $alice $bob --from $alice

$ docker exec -it checkers \

checkersd tx checkers play-move 3 1 2 2 3 --from $alice

$ docker exec -it checkers \

checkersd tx checkers play-move 3 0 5 1 4 --from $bob

Space each tx command from a given account by a couple of seconds so that they each go into a different block - by default checkersd is limited because it uses the account's transaction sequence number by fetching it from the current state.



If you want to overcome this limitation, look at checkersd's --sequence flag:

**Local**

**Docker**



Copy

$ checkersd tx checkers create-game --help

Copy

$ docker exec -it checkers \

checkersd tx checkers create-game --help

And at your account's current sequence. For instance:

**Local**

**Docker**



Copy

$ checkersd query account $alice --output json | jq -r '.sequence'

Copy

$ docker exec -it checkers \

bash -c "checkersd query account $alice --output json | jq -r '.sequence'"

Which returns something like:



Copy

9

With three games in, confirm that you see them all:

**Local**

**Docker**



Copy

$ checkersd query checkers list-stored-game

Copy

$ docker exec -it checkers \

checkersd query checkers list-stored-game

List them again after two, three, four, and five minutes. You should see games 1 and 2 disappear, and game 3 being forfeited by Alice, i.e. red Bob wins:

**Local**

**Docker**



Copy

$ checkersd query checkers show-stored-game 3 --output json | jq '.storedGame.winner'

Copy

$ docker exec -it checkers \

bash -c "checkersd query checkers show-stored-game 3 --output json | jq '.storedGame.winner'"

This prints:



Copy

"r"

Confirm that the FIFO no longer references the removed games nor the forfeited game:

**Local**

**Docker**



Copy

$ checkersd query checkers show-system-info

Copy

$ docker exec -it checkers \

checkersd query checkers show-system-info

This should show:



Copy

SystemInfo:

fifoHeadIndex: "-1"

fifoTailIndex: "-1"

nextId: "4"

synopsis

To summarize, this section has explored:

* How games can expire under two conditions: when the game never really begins or only one player makes an opening move, in which case it is removed; or when both players have participated but one has since failed to play a move in time, in which case the game is forfeited.
* What new information and functions need to be created, and to update EndBlock to call the ForfeitExpiredGames function at the end of each block.
* The correct coding for how to prepare the main loop through the FIFO, identify an expired game, and handle an expired game.
* How to test your code to ensure that it functions as desired.
* How to interact with the CLI to check the effectiveness of your code for handling expired games.

previous

[](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/3-game-fifo.html)

**[Put Your Games in Order](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/3-game-fifo.html)**

up next

**[Let Players Set a Wager](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/5-game-wager.html)**

[[](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/5-game-wager.html)](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/5-game-wager.html)

Rate this Page

icon smile

icon meh

icon frown

Would you like to add a message?

Submit

Thank you for your Feedback!

[](https://ida.interchain.io/ida-course/discord-info.html)

On this page

[New information](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/4-game-forfeit.html#new-information)

[Putting callbacks in place](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/4-game-forfeit.html#putting-callbacks-in-place)

[Expire games handler](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/4-game-forfeit.html#expire-games-handler)

[Prepare the main loop](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/4-game-forfeit.html#prepare-the-main-loop)

[Identify an expired game](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/4-game-forfeit.html#identify-an-expired-game)

[Handle an expired game](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/4-game-forfeit.html#handle-an-expired-game)

[Unit tests](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/4-game-forfeit.html#unit-tests)

[Interact via the CLI](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/4-game-forfeit.html#interact-via-the-cli)

#### **Get Cosmos updates**

Unsubscribe at any time. [Privacy Policy↗](https://v1.cosmos.network/privacy)

     Next

Documentation

[Cosmos SDK](https://docs.cosmos.network/)[Cosmos Hub](https://hub.cosmos.network/)[CometBFT](https://docs.cometbft.com/)[IBC Protocol](https://ibc.cosmos.network/)

Community

[Interchain blog](https://blog.cosmos.network/)[Forum](https://forum.cosmos.network/)[Discord](https://discord.gg/cosmosnetwork)

Contributing

[Source code on GitHub](https://github.com/cosmos/sdk-tutorials)

[](https://ida.interchain.io/)

[Interchain Developer Academy](https://ida.interchain.io/)

**[](https://blog.cosmos.network/)[](https://twitter.com/cosmos)[](https://discord.gg/cosmosnetwork)[](https://www.linkedin.com/company/interchain-foundation/about/)[](https://reddit.com/r/cosmosnetwork)[](https://t.me/cosmosproject)[](https://www.youtube.com/c/CosmosProject)**



Dark mode

† This website is maintained by the Interchain Foundation (ICF). The contents and opinions of this website are those of the ICF. The ICF provides links to cryptocurrency exchanges as a service to the public. The ICF does not warrant that the information provided by these websites is correct, complete, and up-to-date. The ICF is not responsible for their content and expressly rejects any liability for damages of any kind resulting from the use, reference to, or reliance on any information contained within these websites.

Cosmos is a registered trademark of the [Interchain Foundation.](https://interchain.io/)[Privacy](https://v1.cosmos.network/privacy)