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

[Interchain Developer Academy](https://ida.interchain.io/)/[Interchain Developer Academy](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/8-game-winner.html)



Search

[Interchain Developer Academy](https://ida.interchain.io/)[Interchain Developer Academy](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/8-game-winner.html)

Search



Filters

Interchain Developer Academy

[](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/8-game-winner.html)

[Week 0 - Getting Started](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/8-game-winner.html)

[Getting Started](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/8-game-winner.html)

[Blockchain 101](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/8-game-winner.html)

[Blockchain History](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/8-game-winner.html)

[Public and Managed Blockchains](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/8-game-winner.html)

[Consensus in Distributed Networks](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/8-game-winner.html)

[Cryptography](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/8-game-winner.html)

[Self-Assessment Quiz](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/8-game-winner.html)

[Go Introduction - First Steps](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/8-game-winner.html)

[Go Basics](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/8-game-winner.html)

[Go Interfaces](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/8-game-winner.html)

[Control Structures in Go](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/8-game-winner.html)

[Arrays and Slices in Go](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/8-game-winner.html)

[Standard Packages in Go](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/8-game-winner.html)

[Concurrency in Go](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/8-game-winner.html)

[Good-To-Know Dev Terms](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/8-game-winner.html)

[Docker Introduction](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/8-game-winner.html)

[](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/8-game-winner.html)

[Week 1 - Introduction to the Interchain](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/8-game-winner.html)

[Introduction to the Interchain](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/8-game-winner.html)

[Blockchain Technology and the Interchain](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/8-game-winner.html)

[The Interchain Ecosystem](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/8-game-winner.html)

[Getting ATOM and Staking It](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/8-game-winner.html)

[A Blockchain App Architecture](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/8-game-winner.html)

[Accounts](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/8-game-winner.html)

[Transactions](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/8-game-winner.html)

[Messages](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/8-game-winner.html)

[Modules](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/8-game-winner.html)

[Protobuf](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/8-game-winner.html)

[Multistore and Keepers](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/8-game-winner.html)

[BaseApp](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/8-game-winner.html)

[Queries](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/8-game-winner.html)

[Events](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/8-game-winner.html)

[Context](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/8-game-winner.html)

[Testing](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/8-game-winner.html)

[Relaying with IBC](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/8-game-winner.html)

[Interchain Security](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/8-game-winner.html)

[Bridges](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/8-game-winner.html)

[Migrations](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/8-game-winner.html)

[Week 1 Quiz](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/8-game-winner.html)

[](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/8-game-winner.html)

[Week 2 - First Steps](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/8-game-winner.html)

[First Steps](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/8-game-winner.html)

[Setup Your Work Environment](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/8-game-winner.html)

[Run a Node, API, and CLI](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/8-game-winner.html)

[Ignite CLI](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/8-game-winner.html)

[Exercise - Make a Checkers Blockchain](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/8-game-winner.html)

[Store Object](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/8-game-winner.html)

[Create Custom Messages](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/8-game-winner.html)

[Create and Save a Game Properly](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/8-game-winner.html)

[Add a Way to Make a Move](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/8-game-winner.html)

[Emit Game Information](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/8-game-winner.html)

[Record the Game Winner](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/8-game-winner.html)

[Week 2 Exercise](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/8-game-winner.html)

[](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/8-game-winner.html)

[Week 3 - Introduction to IBC and CosmJS](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/8-game-winner.html)

[Introduction to IBC and CosmJS](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/8-game-winner.html)

[What is IBC?](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/8-game-winner.html)

[IBC/TAO - Connections (OPTIONAL)](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/8-game-winner.html)

[IBC/TAO - Channels (OPTIONAL)](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/8-game-winner.html)

[IBC/TAO - Clients (OPTIONAL)](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/8-game-winner.html)

[IBC Token Transfer](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/8-game-winner.html)

[Interchain Accounts (OPTIONAL)](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/8-game-winner.html)

[IBC Middleware (OPTIONAL)](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/8-game-winner.html)

[Create a Custom IBC Middleware (OPTIONAL)](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/8-game-winner.html)

[Integrate IBC Middleware Into a Chain (OPTIONAL)](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/8-game-winner.html)

[IBC Tooling](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/8-game-winner.html)

[What is CosmJS?](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/8-game-winner.html)

[Your First CosmJS Actions](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/8-game-winner.html)

[Compose Complex Transactions](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/8-game-winner.html)

[Learn to Integrate Keplr](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/8-game-winner.html)

[Create Custom CosmJS Interfaces](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/8-game-winner.html)

[](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/8-game-winner.html)

[Week 4 - Ignite CLI and IBC Advanced](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/8-game-winner.html)

[Ignite CLI and IBC Advanced](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/8-game-winner.html)

[Keep an Up-To-Date Game Deadline](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/8-game-winner.html)

[Keep Track Of How Many Moves Have Been Played](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/8-game-winner.html)

[Put Your Games in Order](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/8-game-winner.html)

[Auto-Expiring Games](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/8-game-winner.html)

[Let Players Set a Wager](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/8-game-winner.html)

[Handle wager payments](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/8-game-winner.html)

[Integration tests](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/8-game-winner.html)

[Incentivize Players](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/8-game-winner.html)

[Help Find a Correct Move](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/8-game-winner.html)

[Play With Cross-Chain Tokens](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/8-game-winner.html)

[Understand IBC Denoms](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/8-game-winner.html)

[Go Relayer](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/8-game-winner.html)

[Hermes Relayer](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/8-game-winner.html)

[](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/8-game-winner.html)

[Week 5 - CosmJS Advanced](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/8-game-winner.html)

[CosmJS Advanced](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/8-game-winner.html)

[Create Custom Objects](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/8-game-winner.html)

[Create Custom Messages](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/8-game-winner.html)

[Get an External GUI](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/8-game-winner.html)

[Integrate CosmJS and Keplr](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/8-game-winner.html)

[Backend Script for Game Indexing](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/8-game-winner.html)

[](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/8-game-winner.html)

[Week 6 - IBC Deep Dive](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/8-game-winner.html)

[IBC Deep Dive](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/8-game-winner.html)

[IBC Application Developer Introduction](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/8-game-winner.html)

[Make a Module IBC-Enabled](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/8-game-winner.html)

[Adding Packet and Acknowledgment Data](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/8-game-winner.html)

[Extend the Checkers Game With a Leaderboard](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/8-game-winner.html)

[Create a Leaderboard Chain](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/8-game-winner.html)

[](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/8-game-winner.html)

[Week 7 - From Code to MVP to Production and Migrations](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/8-game-winner.html)

[From Code to MVP to Production and Migrations](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/8-game-winner.html)

[Run in Production](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/8-game-winner.html)

[Prepare the Software to Run](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/8-game-winner.html)

[Prepare a Validator and Keys](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/8-game-winner.html)

[Prepare Where the Node Starts](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/8-game-winner.html)

[Prepare and Connect to Other Nodes](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/8-game-winner.html)

[Configure, Run, and Set Up a Service](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/8-game-winner.html)

[Prepare and Do Migrations](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/8-game-winner.html)

[Simulate Production in Docker](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/8-game-winner.html)

[Tally Player Info After Production](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/8-game-winner.html)

[Add a Leaderboard as a Module](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/8-game-winner.html)

[Migrate the Leaderboard Module After Production](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/8-game-winner.html)

[Simulate a Migration in Docker](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/8-game-winner.html)

[Final Exam](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/8-game-winner.html)

[](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/8-game-winner.html)

[What's Next?](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/8-game-winner.html)

[Continue Your Interchain Journey](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/8-game-winner.html)

Docs Version Switcher

On this page

[New information](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/8-game-winner.html#new-information)

[Update and check for the winner](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/8-game-winner.html#update-and-check-for-the-winner)

[Unit tests](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/8-game-winner.html#unit-tests)

[Interact via the CLI](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/8-game-winner.html#interact-via-the-cli)

[#Copy link](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/8-game-winner.html#record-the-game-winner) **Record the Game Winner**



Make sure you have everything you need before proceeding:

* You understand the concepts of [Protobuf](https://ida.interchain.io/academy/2-cosmos-concepts/6-protobuf.html).
* Go is installed.
* You have the checkers blockchain codebase with the new events. If not, follow the [previous steps](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/7-events.html) or check out the [relevant version (opens new window)↗](https://github.com/cosmos/b9-checkers-academy-draft/tree/two-events).



In this section, you will:

* Check for a game winner.
* Extend unit tests.

In a [previous section](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/6-play-game.html) you made it possible to determine the outcome of a game by declaring a winner. You also [emitted it](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/7-events.html).

The next logical step is to also record this information in storage. That will assist you in discerning between games that are current and those that have reached an end by winning. As you will see in subsequent sections about forfeit, there are other ways for a game to finish.

Therefore, a reasonable field to add is for the **winner**. It needs to contain:

* The winner of a game that reaches completion.
* Or the winner *by forfeit* when a game is expired.
* Or a neutral value when the game is active.



In this section a draw is not handled and it would perhaps require yet another value to be saved in *winner*.

It is time to introduce another consideration. When a game has been won, no one else is going to play it. Its board will no longer be updated and is no longer used for any further on-chain decisions. In effect, the board becomes redundant. With a view to keeping a node's storage requirement low, you should delete the board's content but keep the rest of the game's information.

To keep a trace of the last state of the board, you emit it with an event.

[#Copy link](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/8-game-winner.html#new-information) New information

In the StoredGame Protobuf definition file:



Copy

message StoredGame {

...

+ string winner = 6;

}

proto /

checkers /

stored\_game.proto

[View source→](https://github.com/cosmos/b9-checkers-academy-draft/blob/game-winner/proto/checkers/stored_game.proto" \l "L12" \t "_blank)

Have Ignite CLI and Protobuf recompile this file:

**Local**

**Docker**



Copy

$ ignite generate proto-go

Copy

$ docker run --rm -it \

-v $(pwd):/checkers \

-w /checkers \

checkers\_i \

ignite generate proto-go

Add a helper function to get the winner's address, if it exists. A good location is in full\_game.go:



Copy

func (storedGame StoredGame) GetPlayerAddress(color string) (address sdk.AccAddress, found bool, err error) {

black, err := storedGame.GetBlackAddress()

if err != nil {

return nil, false, err

}

red, err := storedGame.GetRedAddress()

if err != nil {

return nil, false, err

}

address, found = map[string]sdk.AccAddress{

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

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

}[color]

return address, found, nil

}

func (storedGame StoredGame) GetWinnerAddress() (address sdk.AccAddress, found bool, err error) {

return storedGame.GetPlayerAddress(storedGame.Winner)

}

x /

checkers /

types /

full\_game.go

[View source→](https://github.com/cosmos/b9-checkers-academy-draft/blob/game-winner/x/checkers/types/full_game.go" \l "L34-L52" \t "_blank)

[#Copy link](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/8-game-winner.html#update-and-check-for-the-winner) Update and check for the winner

This is a two-part update. You set the winner where relevant, but you also introduce new checks so that a game with a winner cannot be acted upon.

Start with a new error that you define as a constant:



Copy

var (

...

+ ErrGameFinished = sdkerrors.Register(ModuleName, 1110, "game is already finished")

)

x /

checkers /

types /

errors.go

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

And a new event attribute:



Copy

const (

MovePlayedEventType = "move-played"

...

MovePlayedEventWinner = "winner"

+ MovePlayedEventBoard = "board"

)

x /

checkers /

types /

keys.go

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

At creation, in the *create game* message handler, start with a neutral value:



Copy

...

storedGame := types.StoredGame{

...

+ Winner: rules.PieceStrings[rules.NO\_PLAYER],

}

...

x /

checkers /

keeper /

msg\_server\_create\_game.go

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

With further checks when handling a play in the handler:

1. Check that the game has not finished yet:



Copy

...

+ if storedGame.Winner != rules.PieceStrings[rules.NO\_PLAYER] {

+ return nil, types.ErrGameFinished

+ }

isBlack := storedGame.Black == msg.Creator

...

x /

checkers /

keeper /

msg\_server\_play\_move.go

[View source→](https://github.com/cosmos/b9-checkers-academy-draft/blob/game-winner/x/checkers/keeper/msg_server_play_move.go" \l "L21-L23" \t "_blank)

1. Update the winner field, which [remains neutral (opens new window)↗](https://github.com/batkinson/checkers-go/blob/master/checkers/checkers.go#L165) if there is no winner yet, and adjust the board depending on whether the game is finished or not:



Copy

...

- storedGame.Board = game.String()

+ storedGame.Winner = rules.PieceStrings[game.Winner()]

+

+ lastBoard := game.String()

+ if storedGame.Winner == rules.PieceStrings[rules.NO\_PLAYER] {

+ storedGame.Board = lastBoard

+ } else {

+ storedGame.Board = ""

+ }

...

x /

checkers /

keeper /

msg\_server\_play\_move.go

[View source→](https://github.com/cosmos/b9-checkers-academy-draft/blob/game-winner/x/checkers/keeper/msg_server_play_move.go" \l "L61-L68" \t "_blank)

1. Add the new attribute in the event:



Copy

...

ctx.EventManager().EmitEvent(

sdk.NewEvent(types.MovePlayedEventType,

...

sdk.NewAttribute(types.MovePlayedEventWinner, rules.PieceStrings[game.Winner()]),

+ sdk.NewAttribute(types.MovePlayedEventBoard, lastBoard),

),

)

...

x /

checkers /

keeper /

msg\_server\_play\_move.go

[View source→](https://github.com/cosmos/b9-checkers-academy-draft/blob/game-winner/x/checkers/keeper/msg_server_play_move.go" \l "L80" \t "_blank)

Confirm the code compiles, add unit tests, and you are ready to handle the expiration of games.

[#Copy link](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/8-game-winner.html#unit-tests) Unit tests

Add [tests (opens new window)↗](https://github.com/cosmos/b9-checkers-academy-draft/blob/game-winner/x/checkers/types/full_game_test.go#L100-L174) for your new functions.

You also need to update your existing tests so that they pass with a new Winner value. Most of your tests need to add this line:



Copy

...

require.EqualValues(t, types.StoredGame{

...

+ Winner: "\*",

}, game1)

...

x /

checkers /

keeper /

msg\_server\_play\_move\_test.go

[View source→](https://github.com/cosmos/b9-checkers-academy-draft/blob/game-winner/x/checkers/keeper/msg_server_play_move_test.go" \l "L107" \t "_blank)

This "\*" means that in your tests no games have reached a conclusion with a winner. Time to fix that. In a dedicated full\_game\_helpers.go file, prepare all the moves that will be played in the test. For convenience, a move will be written as:



Copy

type GameMoveTest struct {

player string

fromX uint64

fromY uint64

toX uint64

toY uint64

}

x /

checkers /

testutil /

full\_game\_helpers.go

[View source→](https://github.com/cosmos/b9-checkers-academy-draft/blob/game-winner/x/checkers/testutil/full_game_helpers.go" \l "L11-L17" \t "_blank)

If you do not want to create a complete game yourself, you can choose this one:



Copy

var (

Game1Moves = []GameMoveTest{

{"b", 1, 2, 2, 3}, // "\*b\*b\*b\*b|b\*b\*b\*b\*|\*\*\*b\*b\*b|\*\*b\*\*\*\*\*|\*\*\*\*\*\*\*\*|r\*r\*r\*r\*|\*r\*r\*r\*r|r\*r\*r\*r\*"

{"r", 0, 5, 1, 4}, // "\*b\*b\*b\*b|b\*b\*b\*b\*|\*\*\*b\*b\*b|\*\*b\*\*\*\*\*|\*r\*\*\*\*\*\*|\*\*r\*r\*r\*|\*r\*r\*r\*r|r\*r\*r\*r\*"

{"b", 2, 3, 0, 5}, // "\*b\*b\*b\*b|b\*b\*b\*b\*|\*\*\*b\*b\*b|\*\*\*\*\*\*\*\*|\*\*\*\*\*\*\*\*|b\*r\*r\*r\*|\*r\*r\*r\*r|r\*r\*r\*r\*"

...

{"r", 3, 6, 2, 5}, // "\*b\*b\*\*\*\*|\*\*b\*b\*\*\*|\*\*\*\*\*b\*\*|\*\*\*\*\*\*\*\*|\*\*\*\*\*\*\*\*|\*\*r\*\*\*\*\*|\*B\*\*\*b\*\*|\*\*\*\*\*\*\*\*"

{"b", 1, 6, 3, 4}, // "\*b\*b\*\*\*\*|\*\*b\*b\*\*\*|\*\*\*\*\*b\*\*|\*\*\*\*\*\*\*\*|\*\*\*B\*\*\*\*|\*\*\*\*\*\*\*\*|\*\*\*\*\*b\*\*|\*\*\*\*\*\*\*\*"

}

)

x /

checkers /

testutil /

full\_game\_helpers.go

[View source→](https://github.com/cosmos/b9-checkers-academy-draft/blob/game-winner/x/checkers/testutil/full_game_helpers.go" \l "L19-L63" \t "_blank)

You may want to add a small function that converts "b" and "r" into their respective player addresses:



Copy

func GetPlayer(color string, black string, red string) string {

if color == "b" {

return black

}

return red

}

x /

checkers /

testutil /

full\_game\_helpers.go

[View source→](https://github.com/cosmos/b9-checkers-academy-draft/blob/game-winner/x/checkers/testutil/full_game_helpers.go" \l "L65-L70" \t "_blank)

And another that applies all the moves. This could become handy if you have multiple games in the future:



Copy

func PlayAllMoves(

t \*testing.T,

msgServer types.MsgServer,

context context.Context,

gameIndex string,

black string,

red string,

moves []GameMoveTest) {

for \_, move := range Game1Moves {

\_, err := msgServer.PlayMove(context, &types.MsgPlayMove{

Creator: GetPlayer(move.player, black, red),

GameIndex: gameIndex,

FromX: move.fromX,

FromY: move.fromY,

ToX: move.toX,

ToY: move.toY,

})

require.Nil(t, err)

}

}

x /

checkers /

testutil /

full\_game\_helpers.go

[View source→](https://github.com/cosmos/b9-checkers-academy-draft/blob/game-winner/x/checkers/testutil/full_game_helpers.go" \l "L72-L91" \t "_blank)

Now, in a new file, create the test that plays all the moves, and checks at the end that the game has been saved with the right winner:



Copy

func TestPlayMoveUpToWinner(t \*testing.T) {

msgServer, keeper, context := setupMsgServerWithOneGameForPlayMove(t)

ctx := sdk.UnwrapSDKContext(context)

testutil.PlayAllMoves(t, msgServer, context, "1", testutil.Game1Moves)

systemInfo, found := keeper.GetSystemInfo(ctx)

require.True(t, found)

require.EqualValues(t, types.SystemInfo{

NextId: 2,

}, systemInfo)

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

require.True(t, found)

require.EqualValues(t, types.StoredGame{

Index: "1",

Board: "",

Turn: "b",

Black: bob,

Red: carol,

Winner: "b",

}, game)

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

require.Len(t, events, 2)

event := events[0]

require.Equal(t, event.Type, "move-played")

require.EqualValues(t, []sdk.Attribute{

{Key: "creator", Value: bob},

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

{Key: "captured-x", Value: "2"},

{Key: "captured-y", Value: "5"},

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

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

}, event.Attributes[(len(testutil.Game1Moves)-1)\*6:])

}

x /

checkers /

keeper /

msg\_server\_play\_move\_winner\_test.go

[View source→](https://github.com/cosmos/b9-checkers-academy-draft/blob/game-winner/x/checkers/keeper/msg_server_play_move_winner_test.go" \l "L12-L46" \t "_blank)



When checking the attributes, it only cares about the last six:



Copy

event.Attributes[(len(testutil.Game1Moves)-1)\*6:])

Feel free to create another game won by the red player.

[#Copy link](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/8-game-winner.html#interact-via-the-cli) Interact via the CLI

If you have created games in an earlier version of the code, you are now in a broken state. You cannot even play the old games because they have .Winner == "" and this will be caught by the if storedGame.Winner != rules.PieceStrings[rules.NO\_PLAYER] test. At some point, look at [migrations](https://ida.interchain.io/hands-on-exercise/4-run-in-prod/2-migration-info.html) to avoid falling into such a situation with a blockchain in production.

Start again:

**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

Do not forget to export alice and bob again, as explained in an [earlier section under "Interact via the CLI"](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/4-create-message.html).

Confirm that there is no winner for a game when it is created:

**Local**

**Docker**



Copy

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

$ checkersd query checkers show-stored-game 1

Copy

$ docker exec -it checkers \

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

$ docker exec -it checkers \

checkersd query checkers show-stored-game 1

This should show:



Copy

...

winner: '\*'

...

And when a player plays:

**Local**

**Docker**



Copy

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

$ checkersd query checkers show-stored-game 1

Copy

$ docker exec -it checkers \

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

$ docker exec -it checkers \

checkersd query checkers show-stored-game 1

This should show:



Copy

...

winner: '\*'

...

Testing with the CLI up to the point where the game is resolved with a rightful winner is better covered by unit tests (as done here) or with a nice GUI. You will be able to partially test this with the CLI in the [next section](https://ida.interchain.io/hands-on-exercise/2-ignite-cli-adv/4-game-forfeit.html), via a forfeit.

synopsis

To summarize, this section has explored:

* How to prepare for terminating games by defining a **winner** field that differentiates between the outright winner of a completed game, the winner by forfeit when a game is expired, or a game which is still active.
* What new information and functions to add and where, including the winner field, helper functions to get any winner's address, a new error for games already finished, and checks for various application actions.
* How to update your tests to check the functionality of your code.
* How interacting via the CLI is partially impeded by any existing test games now being in a broken state due to the absence of a value in the winner field, with recommendations for next actions to take.

previous

[](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/7-events.html)

**[Emit Game Information](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/7-events.html)**

up next

**[Week 2 Exercise](https://ida.interchain.io/ida-course/exercise-week2.html)**

[[](https://ida.interchain.io/ida-course/exercise-week2.html)](https://ida.interchain.io/ida-course/exercise-week2.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/1-ignite-cli/8-game-winner.html#new-information)

[Update and check for the winner](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/8-game-winner.html#update-and-check-for-the-winner)

[Unit tests](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/8-game-winner.html#unit-tests)

[Interact via the CLI](https://ida.interchain.io/hands-on-exercise/1-ignite-cli/8-game-winner.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)