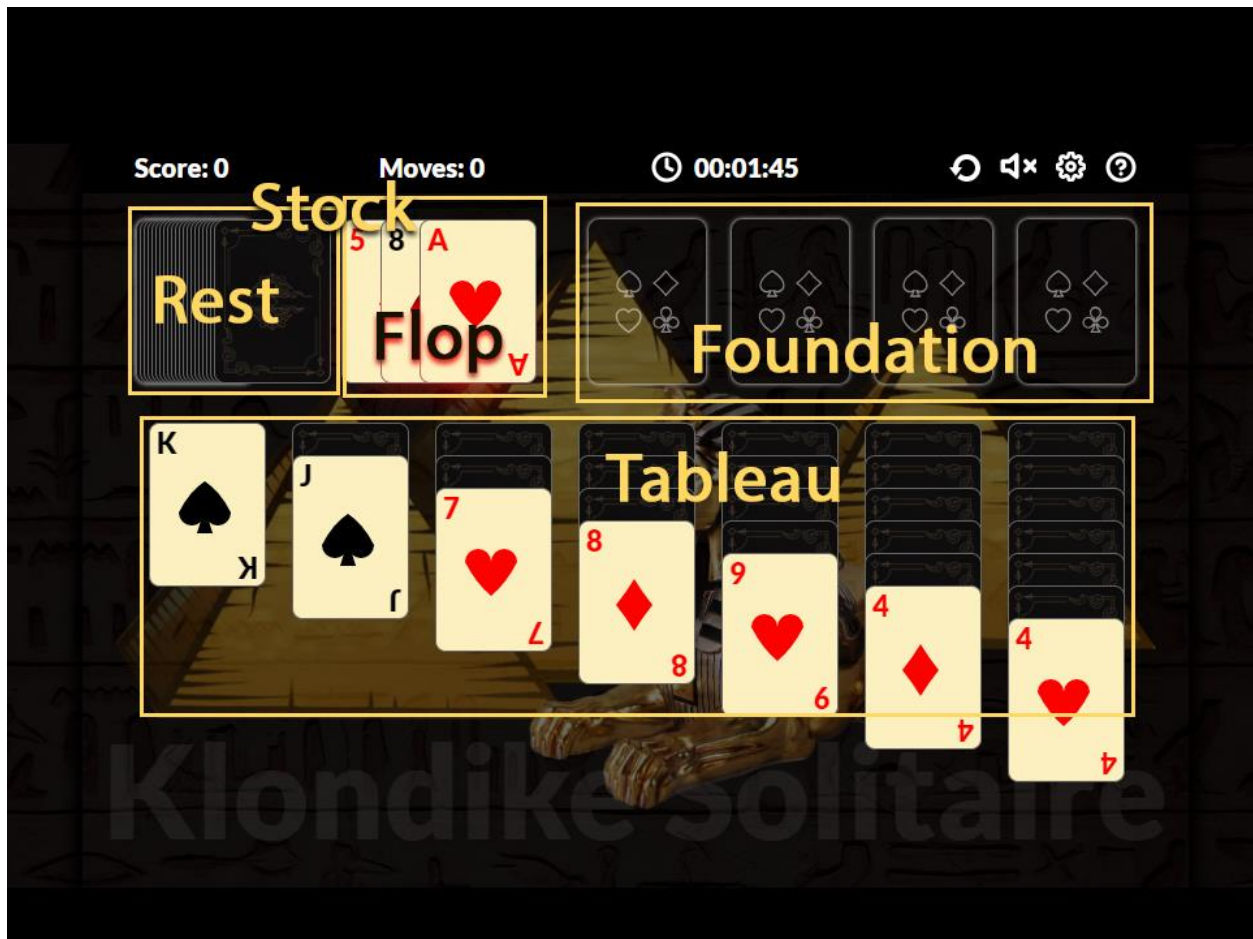# Klondike Solitaire

A very hard but real and proper game play as per the rules of [Klondike Solitaire](#).

How to play?

Please follow the below guide.

From Tableau or from Stock one can put cards starting from A till K into Foundation and for each card user gets 10 points.

From Stock(Flop) one can drag cards and arrange in reverse order from K to 2 in Tableau and user gets 5 points.

With in Tableau one can drag single or multiple cards from one Pile to other to unlock cards underneth.

On card reveal user gets 5 points.

To solve complex situations, user might have to withdraw a card from Foundation to Tableau. In that case user will get negetive 15 points.

The final goal is to move all cards from Rest and Tableau to Foundation.

Configurations:

From Config icon one can any time switch between drawing single card from the Rest into Flop.

User can change sound options.

User can change backgrounds.

How easy is to add more backgrounds or background music?

If you have your own backgrounds and music, just replace them in the assets folder with same names otherwise you might have to change the code.

React Hooks and React's powerful state management?

Yes. The Engine is in React Hooks. React Hooks is the powerful and advanced most popular application development framework. Here I tried to bring the power of the framework to game development.

Why React?

ReactJs and the Hooks concept is the most acceptable and modern frontend technology which grabs more than 73% of the industires user fan base.

Wereever you go you will find ReactJs based applications and developers.

React applications are ready to be imported to React Native very easily which enables anyone to create native applications for IOS and Android.

This application will give you the power of thinking your next game in React Hooks which is even more new trend in the market.

This application uses React Hooks like useReducer, Context API, combine reducers and a very nice application framework with central state management like Redux(Tried to replace Redux).

What about the game framework?

The game is developed by using HTML5, CSS3, JavaScript and Reacts Hooks framework.

What can you expect from us?

We are there to support you through 24x7. For any kind of feature requirements, you can drop a message or a comment. We are ready to help you with in your budget. We are ready to develop multi-player games with this framework for your next big online presence.

What assets I will get?

All high resolution backgrounds

SVG Icons (Few of them are inside code)

Developer guidelines

Complete project with package.json and webpack configuration

Developer Guide

IDE used for the development

1. Visual Studio Code
2. Photoshop for the graphics

Local setup

To setup the development environment locally, please follow the below steps

Front-end: For front end of this game, we use React Hooks framework.

Running the project locally

- Unzip the files,
- From the location of package.json, right click and open integrated terminal in Visual Studio Code or in window address bar, delete the file path and type cmd which opens command line terminal,

- Install all packages for front end by running command `npm i`.
- Then run `npm start` in command line, You are ready to see the app in localhost:3000. If you need to change the port got Webpack.common.js file and modify
  devServer: {
     hot: true,
     port: 3000
   },
- Webpack is used to build the project.
- To make the game engine secure one must obfuscate the worker.js manually using any online obfuscator.

Modifying the theme

To modify the theme or design, one has to go through the code of ReactJs.

Deploying on server

Once your development is ready, you then need to take it to online by hosting your application online in a server. To do this please follow the following steps,

1. Run `npm run build` command from the package.json file location from your terminal.
2. The dist folder is being generated
3. Then this dist folder need to be copied to your server.
4. No need to include any other source code on prod server.