# Setting Up SDL2 with Gloss for Haskell on macOS

This document outlines the steps to set up a Haskell project that uses Gloss for rendering and SDL2 for handling Xbox controller input on macOS. We'll create a new Stack project, add SDL2 dependencies, and implement a basic example that integrates both libraries.

## Step 1: Create a New Stack Project

Open your terminal and run the following commands to create a new Stack project named 'gloss-sdl-example':

```bash  
$ stack new gloss-sdl-example  
$ cd gloss-sdl-example  
```

## Step 2: Modify the stack.yaml File

Open the stack.yaml file and add SDL2 as an extra dependency. Add the following line under 'extra-deps':

```yaml  
extra-deps:  
 - sdl2-2.5.1  
 - sdl2-ttf-0.6.0  
```

## Step 3: Create a New Source File

Open Visual Studio Code by running the following commands:

```bash  
$ cd gloss-sdl-example  
$ code .  
```

Create a new file named `Main.hs` in the `src/` directory with the following content:

```haskell  
module Main where  
  
import Graphics.Gloss  
import Graphics.Gloss.Interface.IO.Game  
import qualified SDL  
import Foreign.C.Types (CInt)  
  
-- Define your game state  
data GameState = GameState  
 { sdlEvents :: [SDL.Event]  
 }  
  
-- Initialize the SDL subsystem  
initializeSDL :: IO ()  
initializeSDL = SDL.initialize [SDL.InitVideo, SDL.InitJoystick]  
  
-- Handle SDL Events  
handleSDLEvents :: [SDL.Event] -> IO ()  
handleSDLEvents events = mapM\_ print events  
  
-- Your Gloss rendering function  
render :: GameState -> Picture  
render \_ = color white $ circleSolid 80  
  
-- Your Gloss event handling function  
handleEvents :: Event -> GameState -> IO GameState  
handleEvents \_ gs = do  
 events <- SDL.pollEvents  
 handleSDLEvents events  
 return gs { sdlEvents = events }  
  
-- The main function  
main :: IO ()  
main = do  
 initializeSDL  
 playIO  
 (InWindow "Gloss SDL Example" (800, 600) (100, 100))  
 black  
 60  
 (GameState [])  
 (return . render)  
 handleEvents  
 (\\_ world -> return world)  
```

## Step 4: Build and Run the Project

Run the following command to build your project:

```bash  
$ stack build  
```

Finally, run your application with:

```bash  
$ stack exec gloss-sdl-example  
```