

## *Beispielanwendung starten*

- 1) Gitbash oder vergleichbare Kommandozeile öffnen
- 2) Navigieren zu **Build/Scenes**
- 3) Befehl: **electron** .

## *Code neu kompilieren*

- 1) Gitbash oder vergleichbare Kommandozeile öffnen
- 2) Navigieren zu **Source/Scenes**
- 3) Befehl: **tsc**

## *Komponenten einbinden*

- Es wird immer **ein FudgeServer** und **wenigstens ein ClientManager** benötigt!!
- FudgeServer und ClientManager müssen ein UI bekommen, es kann die Vorlage im Ordner „HTML“ als Beispiel herangenommen werden
- Importiert werden die Module mittels „import \* as FudgeNetwork ,path to ModuleCollector.ts“

## *Komponenten erweitern*

- Zusätzliche Funktionen können in den FudgeServer und ClientManager TypeScript Dateien direkt programmiert werden
- Empfehlung: Gewünschte Netzwerkstruktur auswählen und **FudgeServer/ClientManager** in neuer Klasse vererben
  - WICHTIG: In diesem Fall sollte die neue Klasse explizit, nicht als Teil des ModuleCollectors importiert werden. Andernfalls kann es zu circular dependencies und Absturz des Programms kommen
- Überschreiben der originären Funktionen ist nicht empfehlenswert.
  - Am besten überschreiben und mit super() aufrufen, dann Daten weiterverarbeiten, so ist die Funktionsfähigkeit gewährleistet
- Komponenten können nun beliebig und gut strukturiert um Funktionen erweitert werden!