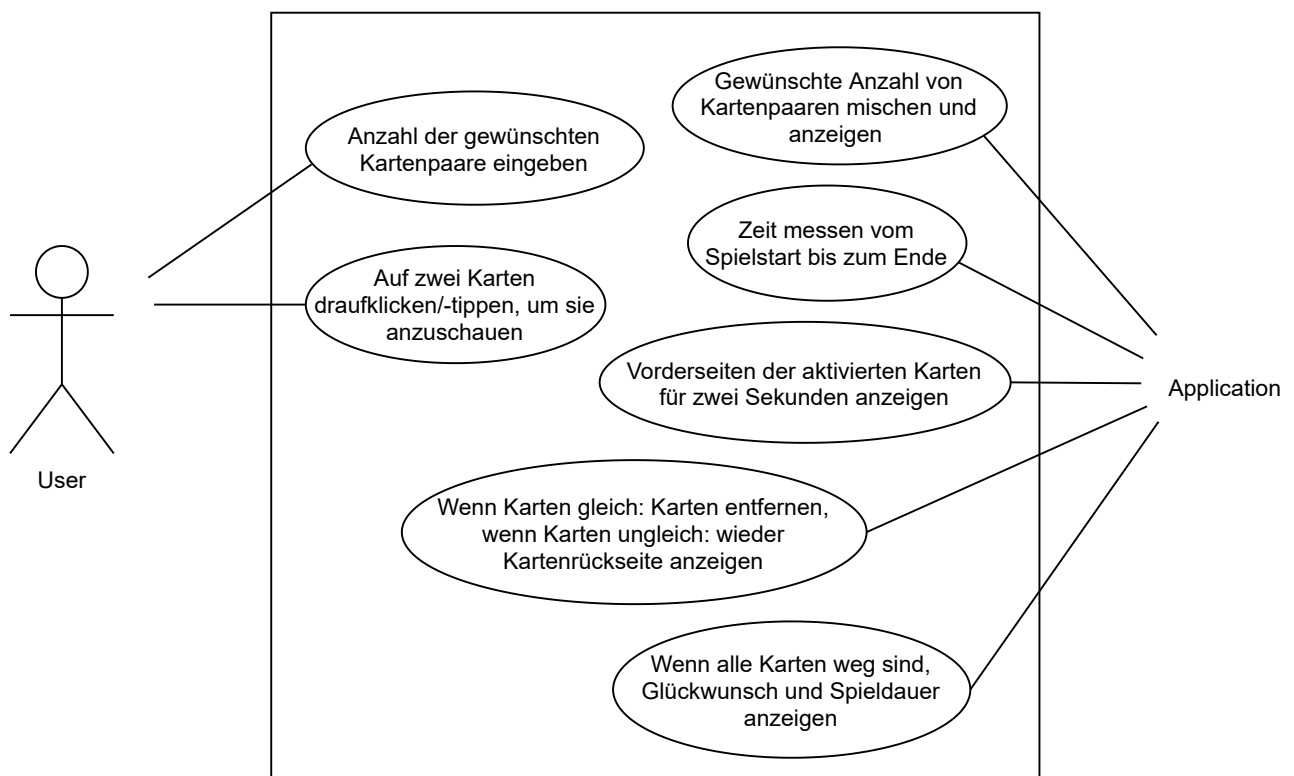
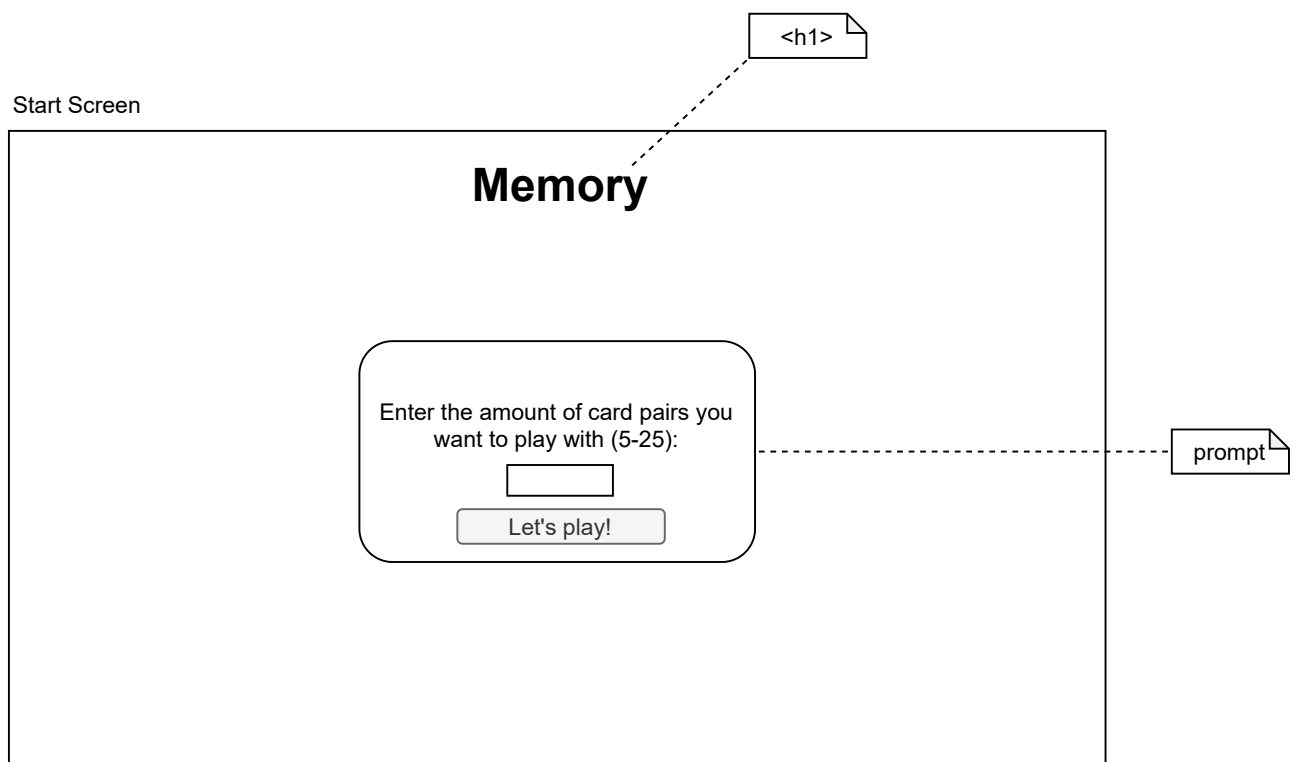
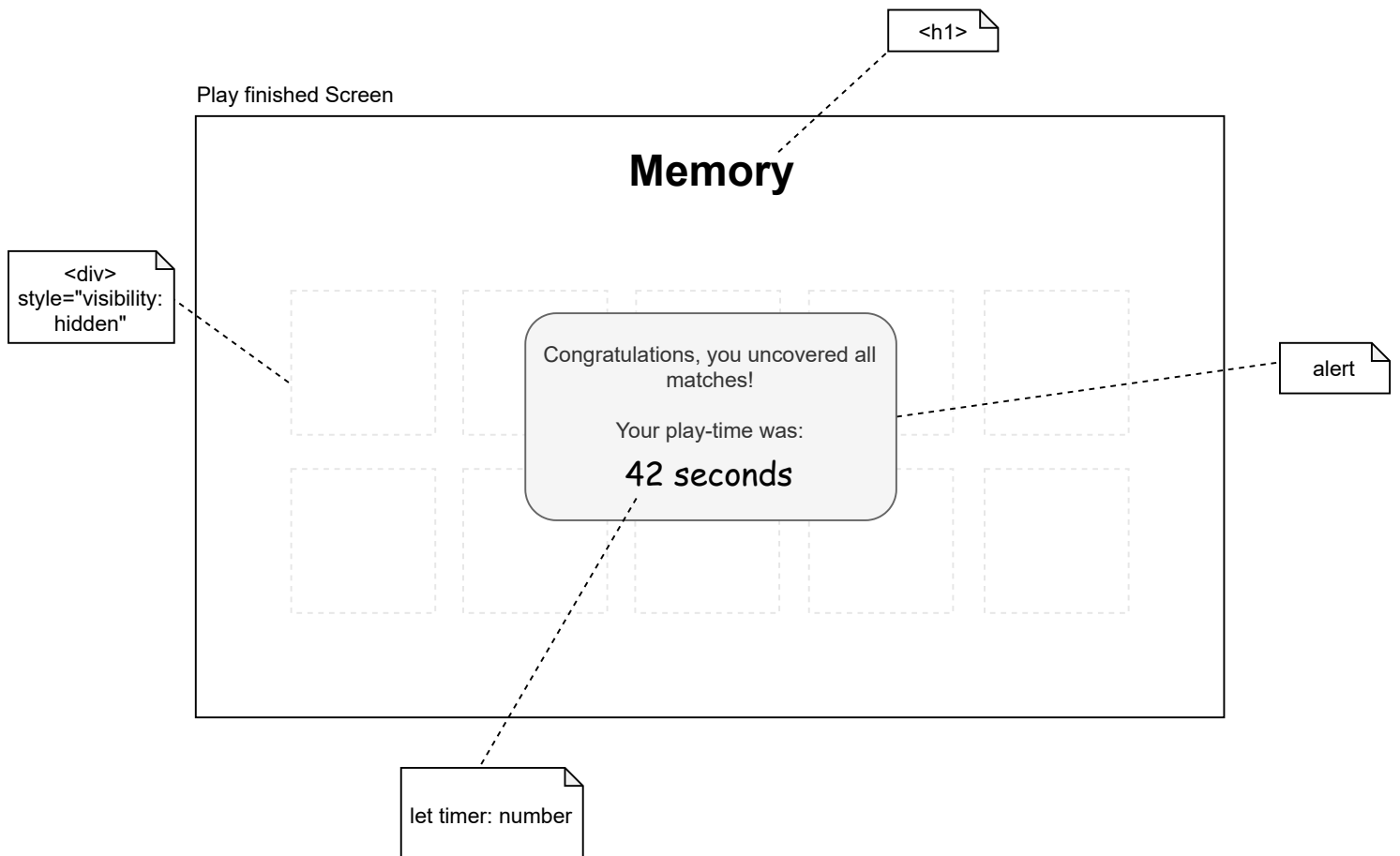
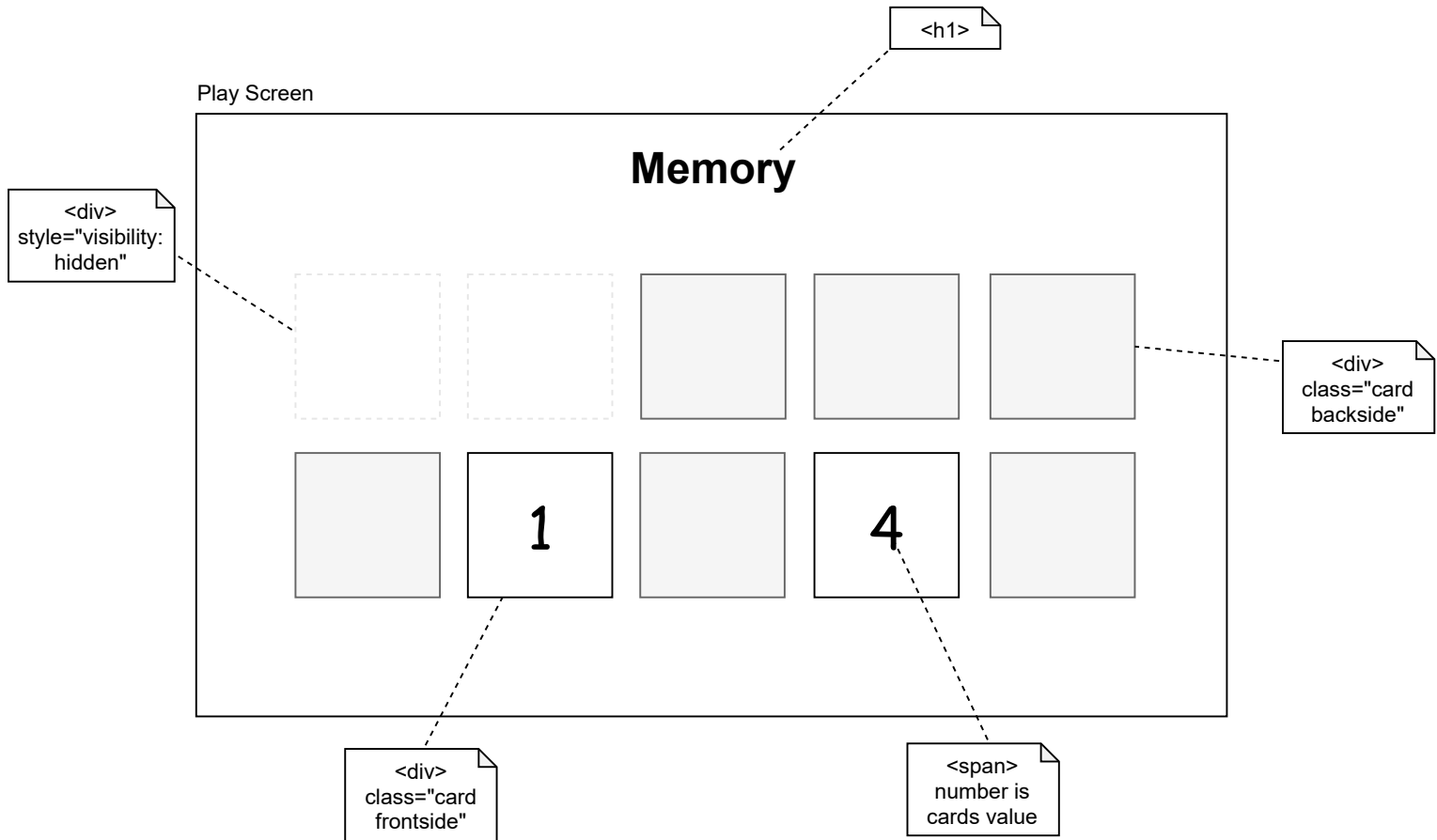


Memory | Use Case Diagram

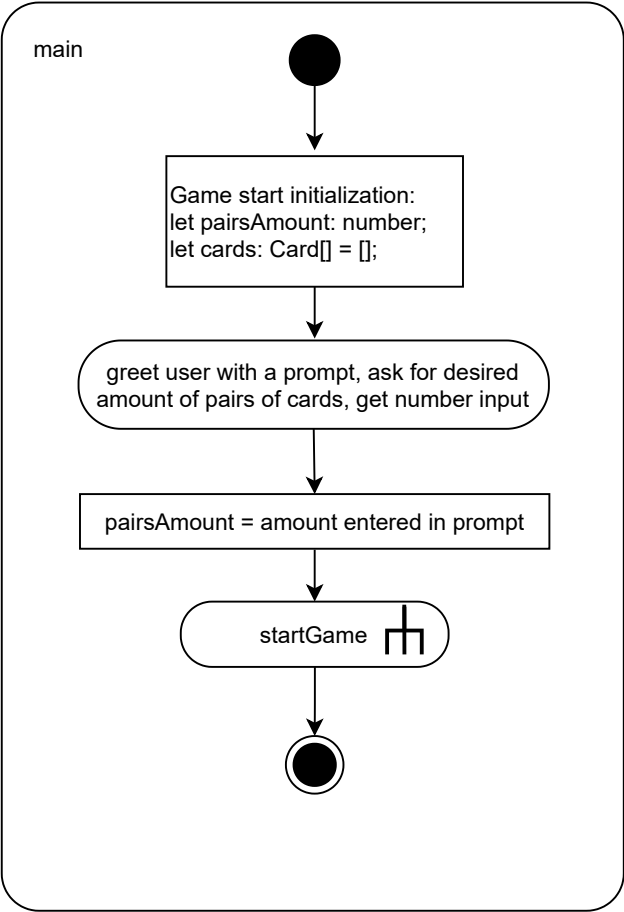


Memory | User Interface Scribble

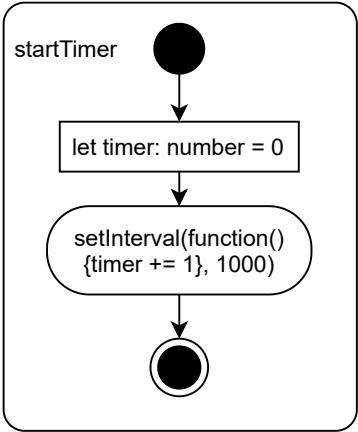
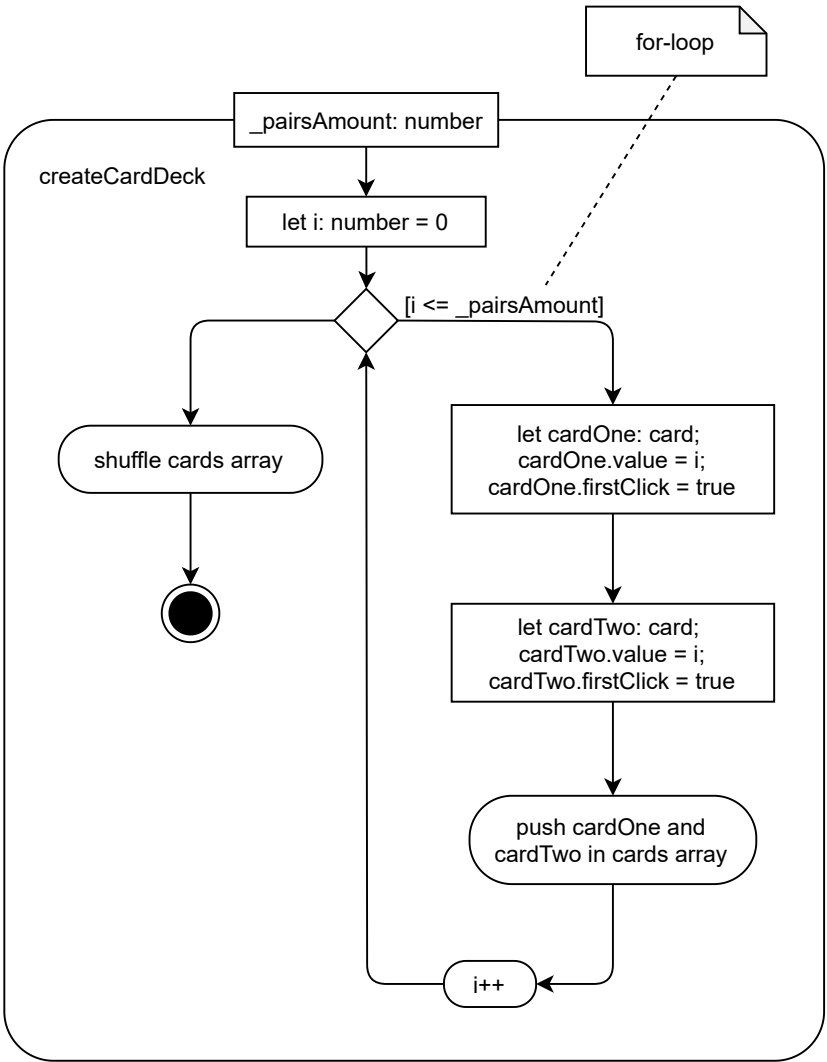
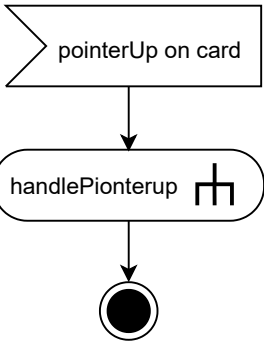
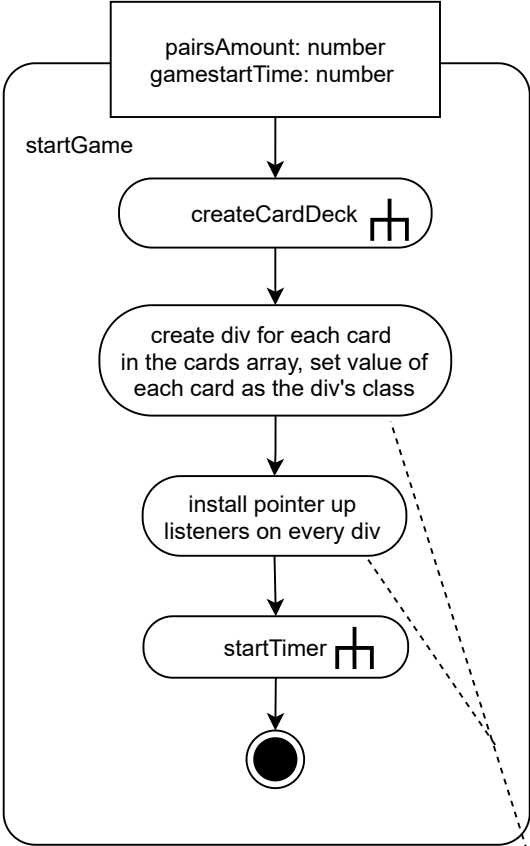


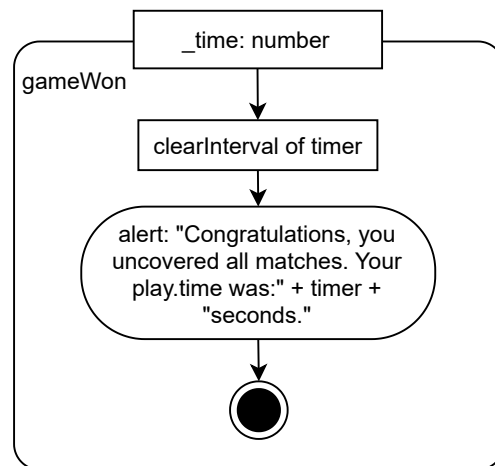
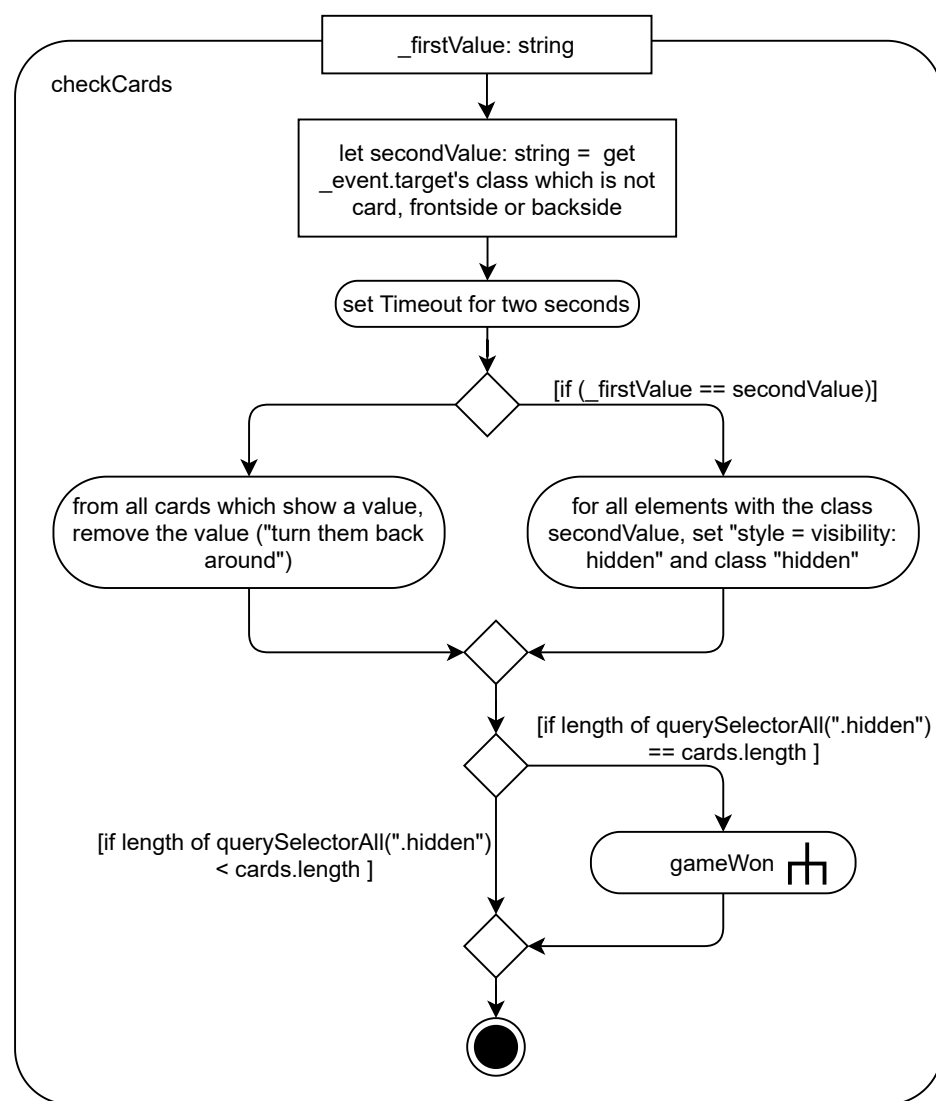
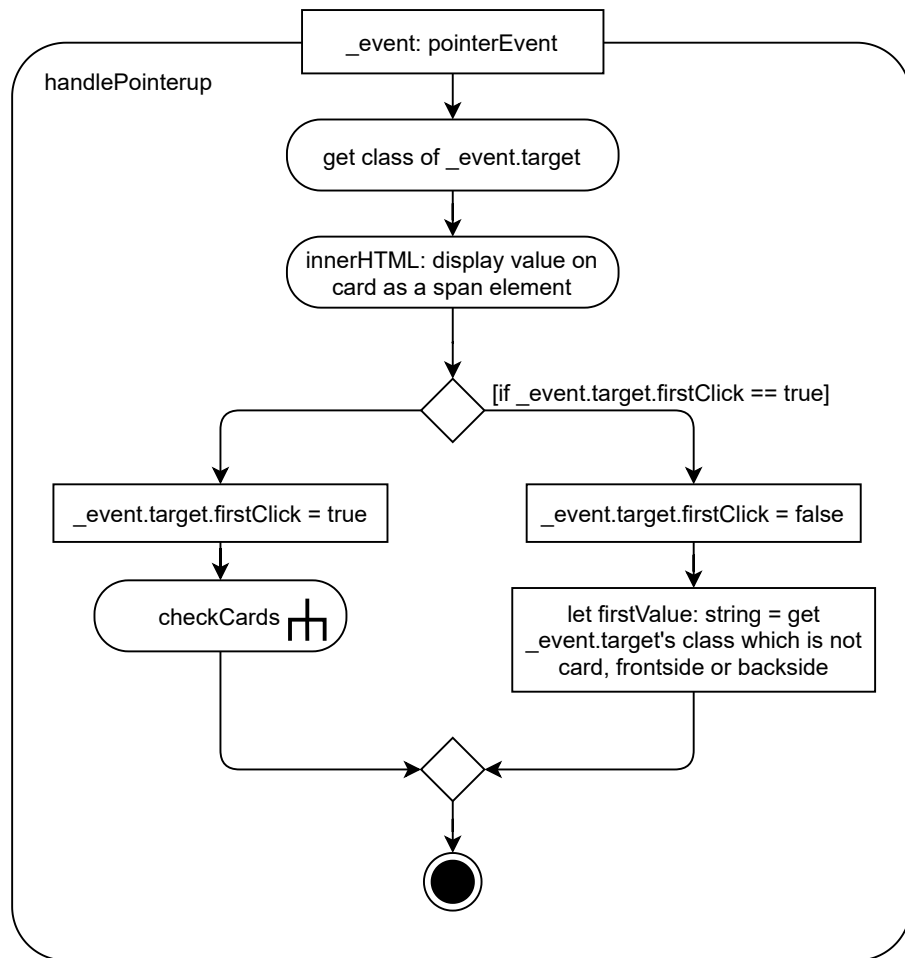


Memory | Activity Diagram



Interface: card
{value: number; firstClick: boolean}





EIAL02 | Memory | Konzept Überarbeitung

Tauschpartner: Sarah Mohring

Punkte, die bei der Korrektur insbesondere anfielen:

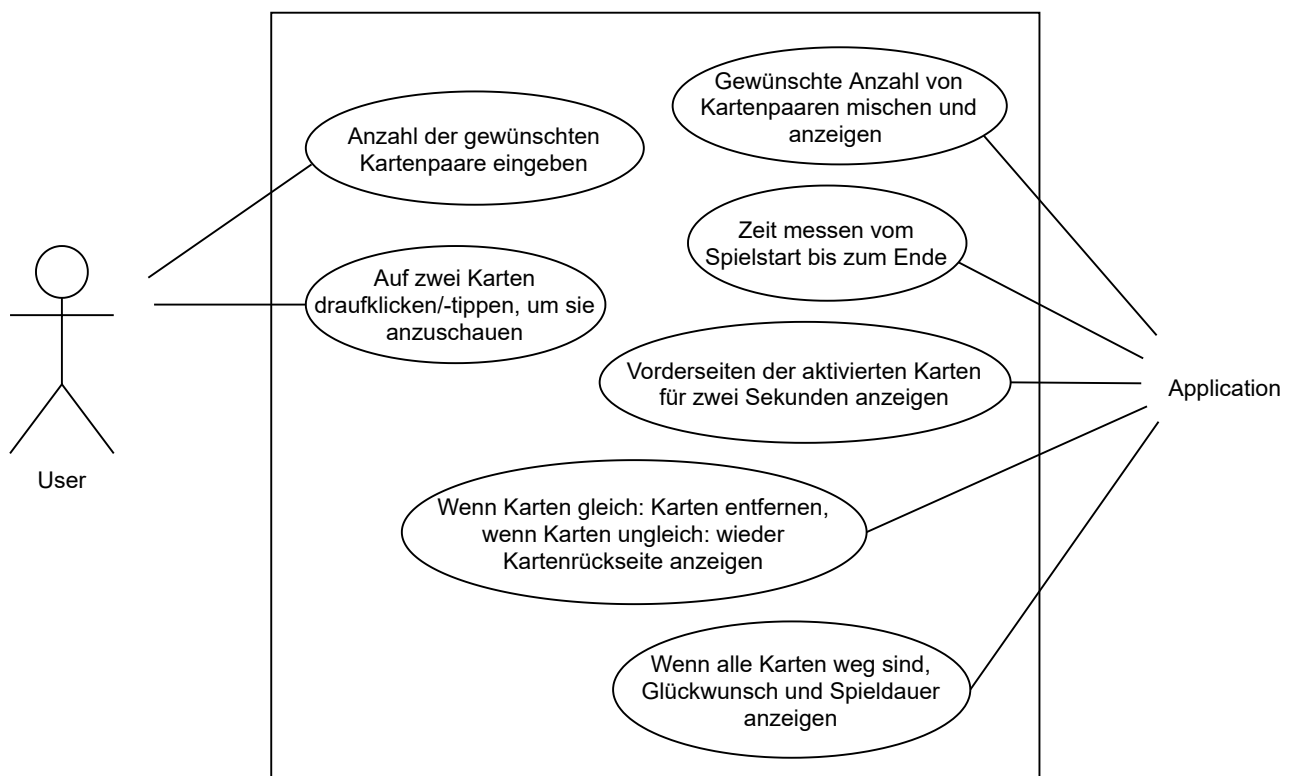
- Insbesondere die **Variablendeklaration**. Oft wurden Variablen mehrfach verwendet, die aber nur in einer Funktion deklariert wurden. Stattdessen werden nun zu Beginn einige global deklariert und ihnen später nur noch Werte zugewiesen.
- **Genauere Ausführung von Aktivitäten**, wie zum Beispiel „shuffle cards“, was in der zweiten Version nun eine eigene Funktion bekommen hat.
- Allgemein das **Durchmischen von verschiedenen Ansätzen** — ich hatte zu Beginn einige Ideen, wie ich Dinge lösen konnte, im Praktikum habe ich dann andere Ansätze mitbekommen, die ich beibehielt. Dadurch entstand ein Mix von beiden Wegen, der manchmal einfach keinen Sinn mehr gemacht hat, z.B. das Umdrehen der Karten mit CSS-Klassen oder durch darstellen des Wertes im Code. Auch der Timer war inkonsequent verwendet.
- Da gar **kein Load-Listener vorhanden** war, hat der Code zuerst überhaupt nicht funktioniert, deswegen wurde in der zweiten Version ein Load-Listener und eine entsprechende Tandler-Funktion eingeführt.
- Im **Aufbau der HTML-Seite** wurde ein Spielfeld-div hinzugefügt, an welches man dann im Code die neu erstellten span-Elemente anhängen konnte.

Anmerkung zur Aufgabenstellung:

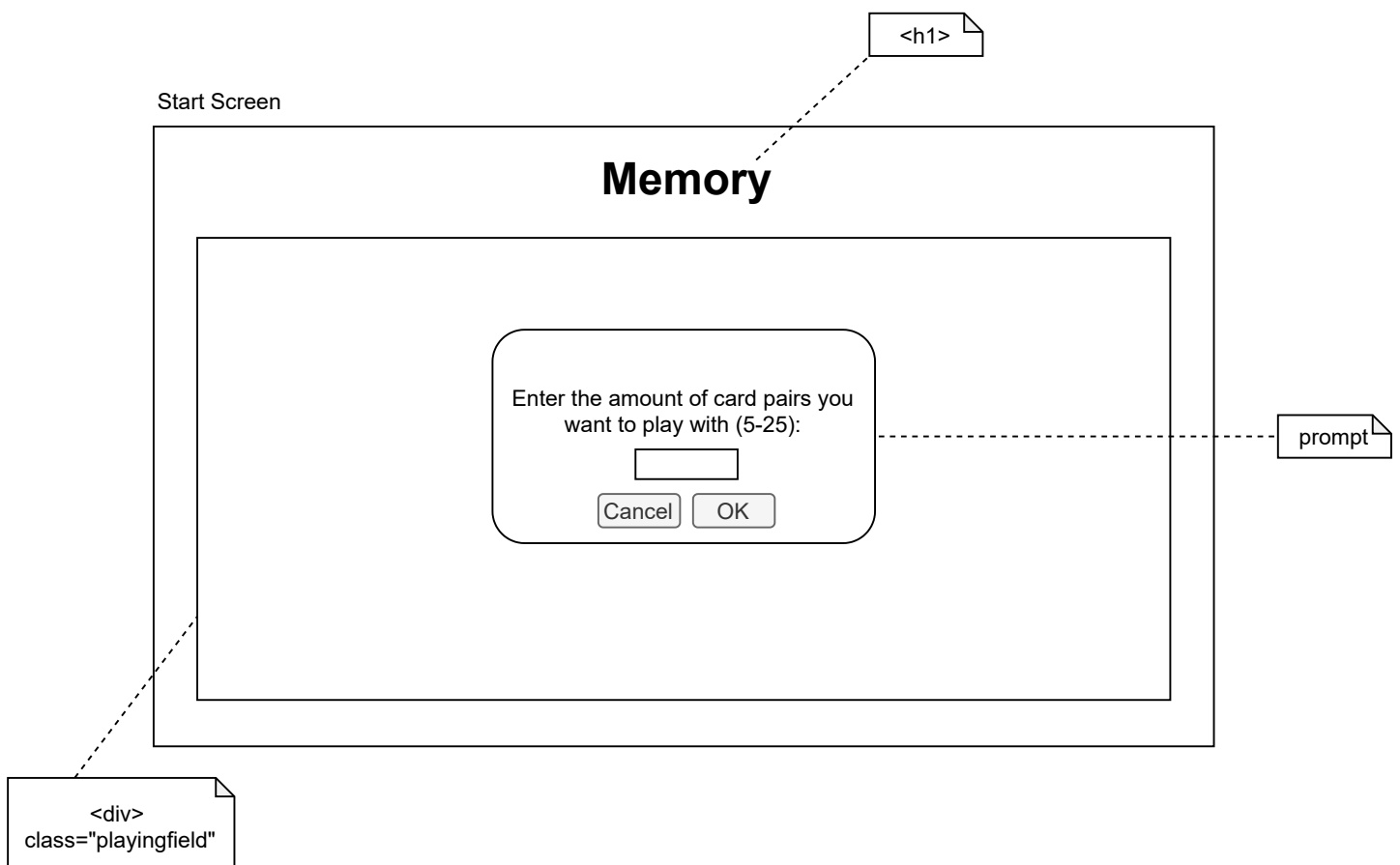
Nachdem man die Konzepte getauscht hat, hat man sich ja mit dem eigenen gar nicht mehr konkret auseinandergesetzt. Stattdessen habe ich mich in Sarahs Konzept reingefuchst, bis ich hier für die meisten Probleme Lösungen gefunden hatte, die Ansätze waren aber teilweise recht unterschiedlich. Anschließend war es ziemlich schwer, wieder in die Denkweise des eigenen Konzeptes reinzukommen, zu dem man zusätzlich noch überhaupt keine praktische Umsetzung hatte. Das war ziemlich schwierig.

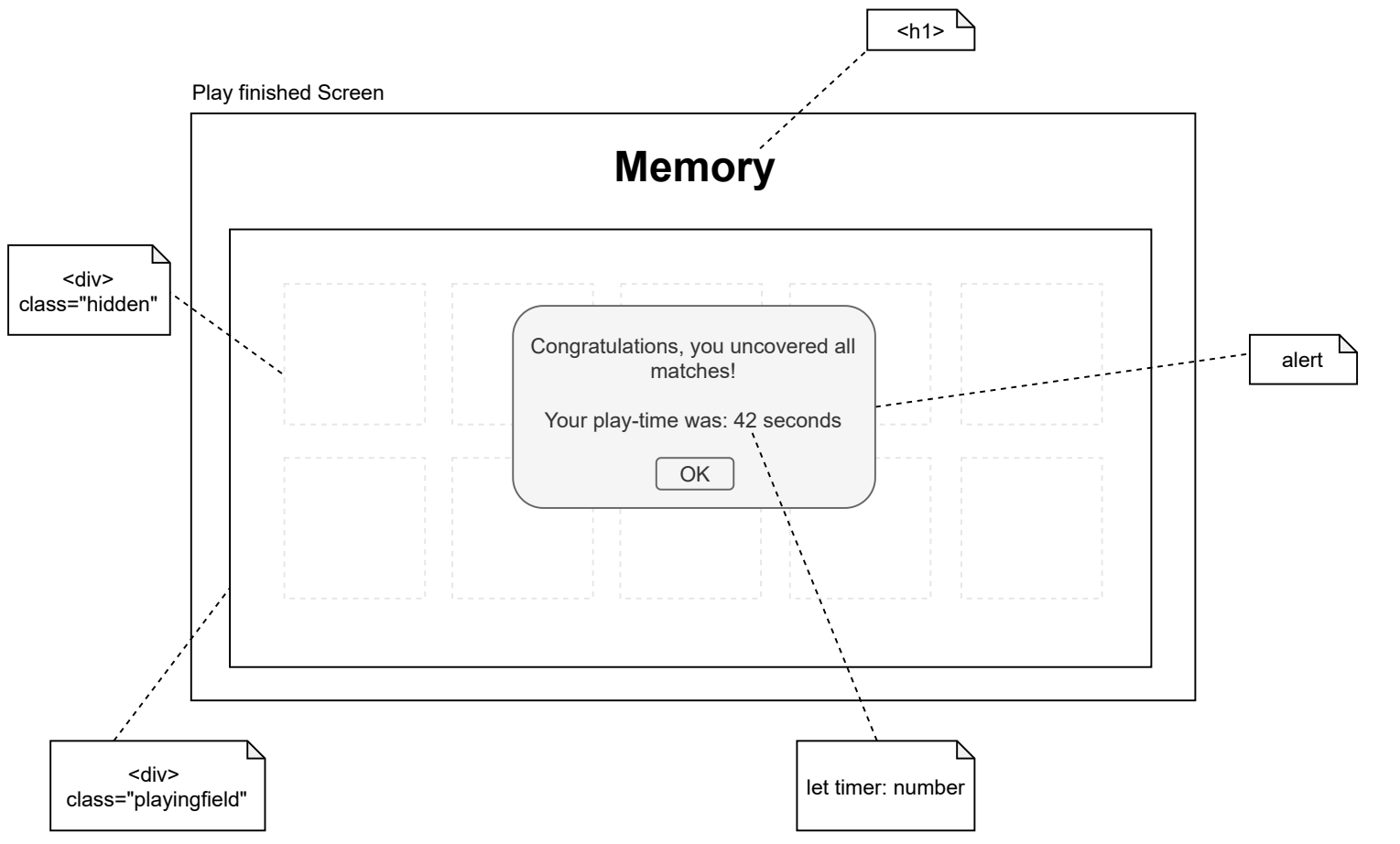
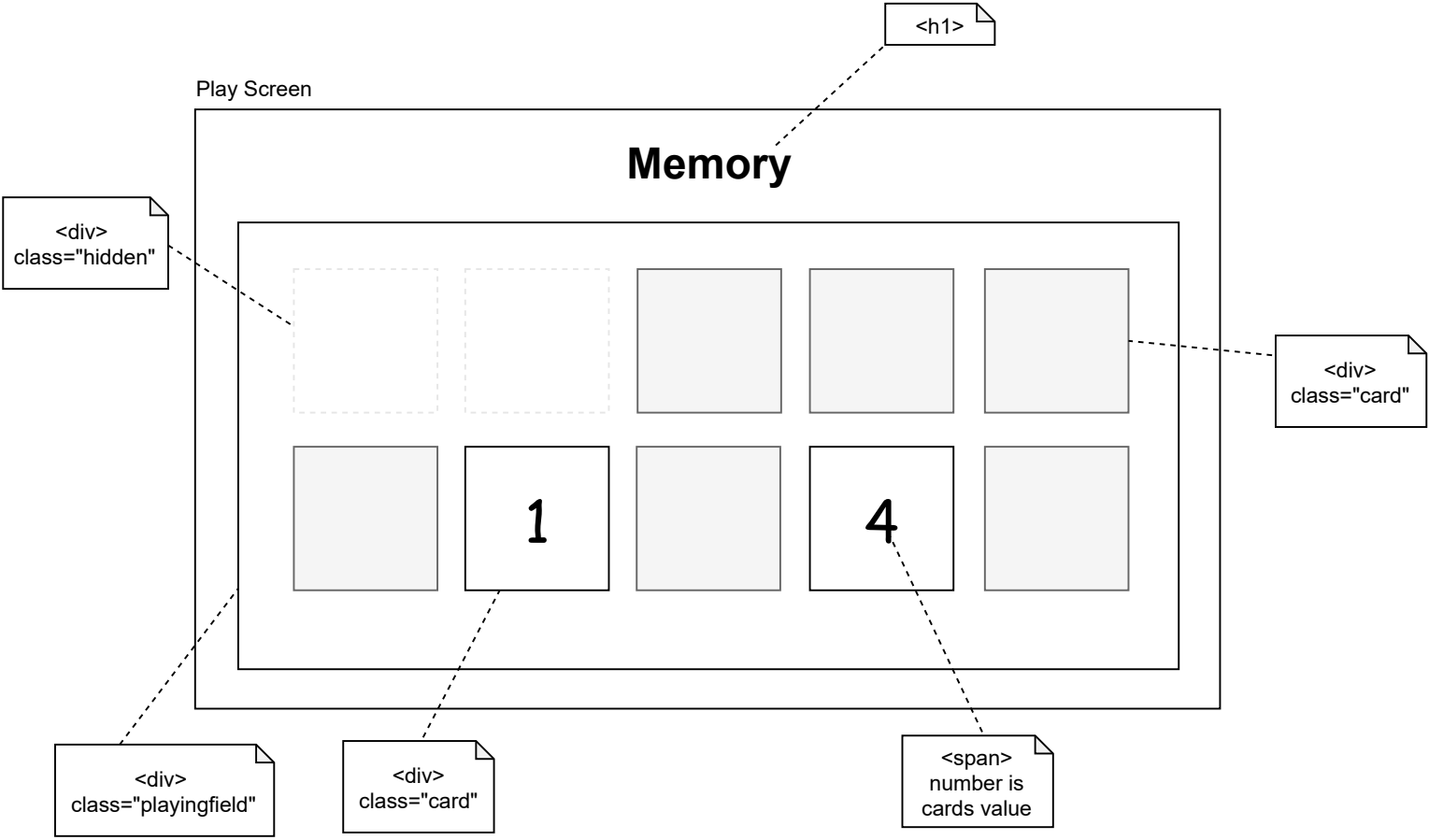
Beide Konzepte zu coden wäre vom Zeitaufwand zu viel gewesen, aber vielleicht das Konzept des Tauschpartners überarbeiten, in das man sich so tief eingearbeitet hat, anstatt das eigene, von dem man sich in der Zwischenzeit gegebenenfalls schon sehr weit entfernt hat?

Memory | Use Case Diagram



Memory | User Interface Scribble





Memory | Activity Diagram

