

LB 426

Ellis Arn & Jan Lehner

Inhalt

Designpatterns.....	2
Entscheidung für das State-Pattern	2
Einsatz des State-Pattern	2
Zusätzliches Pattern Singleton	3
Verwendung des Singleton-Patterns	3
Userstories.....	3
Aufteilung	3
Realisierung auf den Branches	4
Information zu den Screenshots.....	4
Dokumentation der Branches.....	4
UserStory_01	4
UserStory_02	4
UserStory_03	5
UserStory_04	5
UserStory_05	5
UserStory_06	6
UserStory_07	6
UserStory_08	6
UserStory_09	7
UserStory_10	7
Alle Merge requests.....	7
Alle Branches	8
SOLID	8
Single-Responsibility-Prinzip.....	8
Open-Closed Prinzip	8
Liskovsches Substitutionsprinzip	8
Interface Segregation Prinzip.....	8
Dependency-Inversion-Prinzip.....	8
Retrospektive	9
Hindernis 1: Missverständnisse	9
Lösungsvorschlag.....	9
Hindernis 2: Pausen	9
Lösungsvorschlag.....	9
Hindernis 3: Den Wald vor lauter Bäumen nicht sehen	9
Lösungsvorschlag.....	9

Designpatterns

Wie in der Leistungsbeurteilung vorgegeben haben wir das Observer-Pattern mit in unsere Anwendung eingebaut, für das zweite Pattern haben wir uns für das State-Pattern entschieden, wobei es dafür mehrere Gründe gibt.

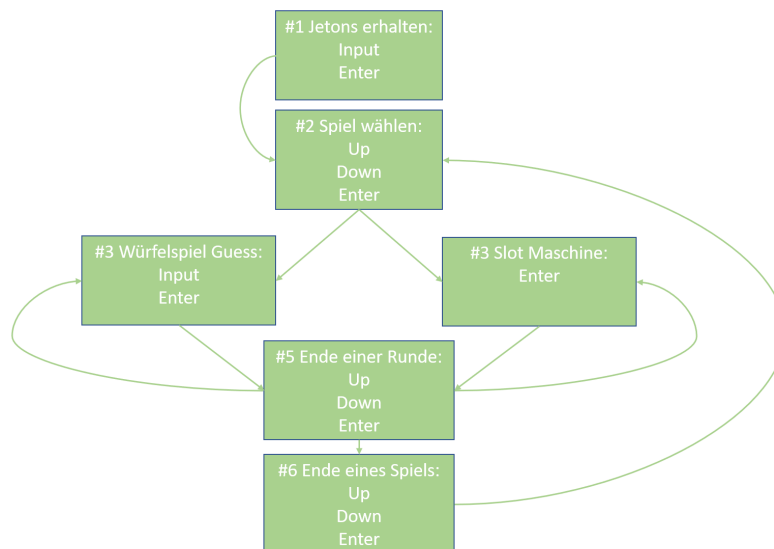
Begründung für das State-Pattern

Der erste Grund, wieso wir das State-Pattern gewählt haben, ist wahrscheinlich auch der wichtigste. In unserer Anwendung wollten wir Benutzereingaben minimieren und daher nur die Hoch- und Runter-Pfeiltasten sowie die Entertaste für die Navigation im Spiel erlauben. Da so eine Taste je nach Stand des Spiels eine andere Funktion haben muss, lag die Idee zu den States sehr nahe. Indem wir das State-Pattern in unserer Anwendung verwenden, können wir immer die gleichen Tasten verwenden und je nach State haben diese eine andere Funktion.

Der zweite Grund für die Wahl war ein kleines Detail in der Aufgabenstellung. In der LB steht nämlich folgendes: *«Die Software soll so konzipiert sein, dass neue Spiele einfach integriert werden können.»* Genau diese Anforderung war ein weiterer Grund für das State-Pattern, so können wir nämlich ein neues Spiel einfach zu einem State machen und so dieses bei der Auswahl der States hinzufügen.

Einsatz des State-Pattern

Das State-Pattern haben wir in unserem Spiel für die einzelnen Teile des Casinos verwendet, hier eine bildliche Übersicht dazu:



In der Abbildung sind alle States zu sehen und zusätzlich dazu die Verbindungen unter diesen. Im Programm stehen folgende Klassen direkt mit dem State-Patter im Zusammenhang:

- EndOfGames.cs
- EndOfRound.cs
- GameChooser.cs
- JetonChange.cs
- Menu.cs
- DiceGame.cs
- Slotmachine.cs

Begründung für das Observer Pattern

Abgesehen von der Anforderung, dass wir dieses Pattern benutzen müssen, haben wir es gewählt, da wir die aktuelle Anzahl von Jetons für mehrere Funktionen gebraucht haben. Einmal für die Jeton-Anzeige und das andere Mal für den Security. Mithilfe des Observer-Pattern konnten wir bei jeder Jeton Änderung die Anzeige aktualisieren und prüfen, ob der Security den Spieler rausschmeissen muss (wenn der Spieler keine Jetons mehr hat).

Zusätzliches Pattern Singleton

Zusätzlich zu den Pattern Observer und State haben wir ausserdem das Singleton Pattern in unserer Anwendung benutzt. Wir haben dieses Pattern verwendet, da es uns den Zugriff auf die Instanz der Spielerklasse von überall in der Anwendung ermöglicht. Da nur eine einzige Instanz existiert, können andere Teile des Programms problemlos auf die Jeton-Anzahl des Spielers und die RefreshArea-Methode der TextArea-Klasse zugreifen, ohne die Instanz explizit zu übergeben oder neu zu erstellen.

Verwendung des Singleton-Patterns

Wir haben das Singleton Pattern in den Klassen «Player.cs» und «Textarea.cs» verwendet.

Userstorys

Die Userstory zum Projekt finden Sie in der Datei «ArnLehnerLB-426-UserStorys.pdf».

Aufteilung

Bei der Aufteilung der Userstorys haben wir versucht darauf zu achten, dass beide Mitglieder etwa denselben Umfang erhalten. Folgende Aufteilung ist dabei entstanden:

Ellis Arn:

- #01 Jetons erhalten
- #02 Auswahl der Spiele
- #06 Würfel Spiel
- #09 Jeton-Anzeige
- #10 Text-Anzeige

Jan Lehner:

- #03 Ende einer Runde
- #04 Ende eines Spiels
- #05 Slot Maschine
- #07 Punkte-System
- #08 Security

Die Aufteilung in der richtigen Reihenfolge finden Sie zusätzlich auch noch in der Datei «ArnLehnerLB-426-UserStorys_Aufteilung.pdf».

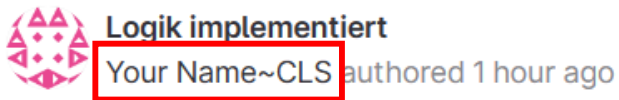
Realisierung auf den Branches

In diesem Kapitel dokumentieren wir die Realisierung des Projekts, welches wir auf dem GitLab der BBB mithilfe von Branches erstellt haben.

[Den Link zum Repository finden Sie hier.](#)

Information zu den Screenshots

Die in diesem Kapitel verwendeten Screenshots wurden von Jan Lehner erstellt, wenn Sie daher in den Commits «Your Name~CLS» sehen ist damit Jan Lehner gemeint:









Dokumentation der Branches

UserStory_01

Commits

UserStory_01 ▾ arn-lehner-lb426 Author ▾ Create merge request Search by message 🔔

21 May, 2023 5 commits

 Logik implementiert Ellis Arn authored 5 hours ago	✓	64371501		
 Tests erstellt Ellis Arn authored 5 hours ago	✗	5a1fe54b		

Merge

User story 01

Edit Code ▾ ⋮







 Merged E.Arn.inf21 requested to merge UserStory_01 into main 5 hours ago

UserStory_02

Commits

UserStory_02 ▾ arn-lehner-lb426 Author ▾ Create merge request Search by message 🔔

21 May, 2023 2 commits

 Logik implementiert Ellis Arn authored 6 hours ago	✓	4b12568c		
 Tests erstellt Ellis Arn authored 6 hours ago	✗	e0f91d61		

Merge

User story 02

Edit Code ▾ ⋮

 Merged E.Arn.inf21 requested to merge UserStory_02 into main 6 hours ago

UserStory_03

Commits

UserStory_03

arn-lehner-lb426

Author

Create merge request

Search by message

20 May, 2023 18 commits

Logik implementiert

Your Name~CLS authored 2 hours ago

ebfa2dc2

Tests erstellt

Your Name~CLS authored 3 hours ago

a615d435

Merge

User story 03

Edit

Code

Merged

J.Lehner.inf21 requested to merge UserStory_03 into main 2 hours ago

Overview

Commits

Pipelines

Changes

UserStory_04

Commits

UserStory_04

arn-lehner-lb426

Author

Create merge request

Search by message

20 May, 2023 24 commits

Logik implementiert

Your Name~CLS authored 1 hour ago

3dc99b58

Tests erstellt

Your Name~CLS authored 1 hour ago

02c98ea5

Merge

User story 04

Edit

Code

Merged

J.Lehner.inf21 requested to merge UserStory_04 into main 1 hour ago

Overview

Commits

Pipelines

Changes

UserStory_05

Commits

UserStory_05

arn-lehner-lb426

Author

Create merge request

Search by message

20 May, 2023 12 commits

Logik implementiert

Your Name~CLS authored 3 hours ago

a2dd129f

Tests erstellt

Your Name~CLS authored 4 hours ago

4ca1aa8d

Merge

User story 05

Edit

Code

Merged

J.Lehner.inf21 requested to merge UserStory_05 into main 3 hours ago

UserStory_06

Commits

UserStory_06 | arn-lehner-lb426

Author | Create merge request | Search by message

20 May, 2023 15 commits

Logik implementiert

Ellis Arn authored 3 hours ago

87e6c85f

87e6c85f

87e6c85f

Tests erstellt

Ellis Arn authored 4 hours ago

65bbd46d

65bbd46d

65bbd46d

Merge

User story 06

Edit

Code

Merged E.Arn.inf21 requested to merge UserStory_06 into main 3 hours ago

UserStory_07

Commits

UserStory_07 | arn-lehner-lb426

Author | Create merge request | Search by message

20 May, 2023 5 commits

Logik implementiert

Your Name~CLS authored 5 hours ago

daf949a0

daf949a0

daf949a0

Tests erstellt

Your Name~CLS authored 5 hours ago

13e02a1b

13e02a1b

13e02a1b

Merge

User story 07

Edit

Code

Merged J.Lehner.inf21 requested to merge UserStory_07 into main 5 hours ago

UserStory_08

Commits

UserStory_08 | arn-lehner-lb426

Author | Create merge request | Search by message

21 May, 2023 8 commits

Logik implementiert

Your Name~CLS authored 2 hours ago

ba0d457a

ba0d457a

ba0d457a

Tests erstellt

Your Name~CLS authored 2 hours ago

d62597b3

d62597b3

d62597b3

Merge

User story 08

Edit

Code

Merged J.Lehner.inf21 requested to merge UserStory_08 into main 2 hours ago

UserStory_09

Commits

UserStory_09


arn-lehner-lb426



Author


Create merge request



Search by message

20 May, 2023 12 commits

 **Logik implementiert**
Ellis Arn authored 4 hours ago

✓ b843bcb5  

 **Tests erstellt**
Ellis Arn authored 4 hours ago

✗ 6b527881  

Merge

User story 09

Edit

Code

 Merged E.Arn.inf21 requested to merge UserStory_09 into main 4 hours ago

UserStory_10

Commits

UserStory_10


arn-lehner-lb426



Author


Create merge request



Search by message

20 May, 2023 6 commits

 **Logik implementiert**
Ellis Arn authored 5 hours ago

✓ d72016c9  

 **Tests erstellt**
Ellis Arn authored 5 hours ago

✗ 65ce2fc5  

Merge

User story 10

Edit

Code

 Merged E.Arn.inf21 requested to merge UserStory_10 into main 5 hours ago

Overview 0 Commits 2 Pipelines 1 Changes 2

Alle Merge requests

J.Lehner.inf21 > ArnLehnerLB-426 > Merge requests

Open 0 Merged 10 Closed 0 All 10

Subscribe to RSS feed

Edit merge requests

New merge request





Recent searches

Search or filter results...

Title





User story 01

!9 · created 5 hours ago by E.Arn.inf21

MERGED ✓    Approved 1  0 updated 5 hours ago





User story 02

!8 · created 6 hours ago by E.Arn.inf21

MERGED ✓    Approved 1  0 updated 6 hours ago





User story 03

!6 · created 19 hours ago by J.Lehner.inf21

MERGED ✓    Approved 2  0 updated 19 hours ago





User story 04

!7 · created 18 hours ago by J.Lehner.inf21

MERGED ✓    Approved 2  0 updated 18 hours ago





User story 05

!4 · created 20 hours ago by J.Lehner.inf21

MERGED ✓    Approved 2  0 updated 20 hours ago





User story 06

!5 · created 20 hours ago by E.Arn.inf21

MERGED ✓    Approved 1  0 updated 20 hours ago





User story 07

!1 · created 21 hours ago by J.Lehner.inf21

MERGED ✓    Approved 2  0 updated 21 hours ago





User story 08

!10 · created 2 hours ago by J.Lehner.inf21

MERGED ✓    Approved 2  0 updated 2 hours ago





User story 09

!3 · created 21 hours ago by E.Arn.inf21

MERGED ✓    Approved 2  0 updated 21 hours ago

User story 10

!2 · created 21 hours ago by E.Arn.inf21

MERGED ✓    Approved 2  0 updated 21 hours ago

Alle Branches

J.Lehner.inf21 > ArnLehnerLB-426 > Repository > Branches

Overview	Active	Stale	All	Filter by branch name	Updated date	Delete merged branches	New branch
main	default	protected					
a0f4685e	Code Smells entfernt (Clean Code), SOLID-Kriterien erfüllt und Tests erweitert/vereinfacht	20 minutes ago					
UserStory_08	merged						
ba0d457a	Logik implementiert	4 hours ago	4 0				
UserStory_01	merged						
64371501	Logik implementiert	8 hours ago	7 0				
UserStory_02	merged						
4b12568c	Logik implementiert	8 hours ago	10 0				
UserStory_04	merged						
3dc99b58	Logik implementiert	20 hours ago	13 0				
UserStory_03	merged						
ebfa2dc2	Logik implementiert	21 hours ago	19 0				
UserStory_06	merged						
87e6c85f	Logik implementiert	22 hours ago	22 0				
UserStory_05	merged						
a2dd129f	Logik implementiert	22 hours ago	25 0				
UserStory_09	merged						
b843bcb5	Logik implementiert	23 hours ago	25 0				
UserStory_10	merged						
d72016c9	Logik implementiert	23 hours ago	31 0				
UserStory_07	merged						
daf949a0	Logik implementiert	1 day ago	32 0				

SOLID

Single-Responsibility-Prinzip

✓ → Erfüllt.

Open-Closed Prinzip

✓ → Erfüllt, siehe Hinzufügen von States.

Liskovsches Substitutionsprinzip

Zunächst haben wir Objekte falsch erstellt und das Prinzip war verletzt. Im Commit «Hier Name» haben wir dieses Problem jedoch behoben.

✓ → Fehler behoben und somit nachträglich erfüllt (siehe Commit: «bda6deb6»).

Interface Segregation Prinzip

✓ → Erfüllt.

Dependency-Inversion-Prinzip

✓ → Erfüllt.

Retrospektive

Hindernis 1: Missverständnisse

Allgemein sind wir mit der Zusammenarbeit zufrieden, jedoch gab es einige Male kleine Missverständnisse. Während dem Projekt haben wir meistens distanziert gearbeitet, also nicht zusammen in einem Raum, sondern über dem Computer in einem Call. Das Problem mit den Missverständnissen können wir darauf zurückführen, da wir während dem Telefonieren etwas gemacht haben und daher nicht 100% auf das Telefonat konzentriert waren.

Lösungsvorschlag

Da wir den Fokus nicht nur auf dem Telefonat, sondern auch auf die Arbeit hatten, kam es zu Missverständnissen. Unsere Idee, beide Teilnehmer sollen die Kamera einschalten. Schalten beide die Kamera ein, fällt es einem einerseits automatisch leichter sich auf das Gegenüber zu konzentrieren, andererseits merkt der Gesprächspartner, wenn man etwas anderes macht und so kann man sich gegenseitig kontrollieren.

Hindernis 2: Pausen

Während dem Realisieren des Projekts haben wir oft viele Stunden am Stück ohne Pause gearbeitet. Man könnte meinen, dass dies gut ist, da man sich so richtig auf das Thema konzentrieren kann und auch auf eine Art eingearbeitet ist. Das Problem dabei war, dass wir mit der Zeit durch das ununterbrochene Starren auf den Bildschirm wir müde wurden und so die Effizienz schwand.

Lösungsvorschlag

Der Lösungsvorschlag zu diesem Problem ist ziemlich eindeutig, Pausen machen! Da diese jedoch schnell vergessen gehen können, sollten wir bei zukünftigen Projekten fixe Zeiten abmachen, an denen wir Pausen machen. Zusätzlich dazu müssen wir einen Wecker stellen, da man in der Arbeit auch gerne mal die Zeit vergisst.

Hindernis 3: Den Wald vor lauter Bäumen nicht sehen

Das letzte Hindernis, welches uns auffiel, ist, zu viel Nachdenken. Immer wieder standen wir vor neuen Dingen, bei denen wir noch nicht so richtig wussten, wie wir es umsetzen sollen. Beide Gruppenmitglieder tendierten dabei zu vielem Nachdenken, wodurch einige Zeit verging.

Lösungsvorschlag

Nachdem wir viel nachgedacht haben, versuchten wir anschliessend einige Dinge und meistens konnte man beim «Testen» schneller Dinge herausfinden. Für zukünftige Projekte sollten wir uns daher ein «Denkfrist» setzen, sich also einen Timer stellen und nach Ablauf dieses einfach probieren und nicht mehr studieren.