|  |
| --- |
| **Use Case:** Schnelles Spiel |
| **Umfang: „**Docker“, Android Anwendung/Spiel  **Ebene:** Anwenderziel  **Primärakteur:** Spieler  **Stakeholder und Interessen:**   * Spieler: Möchte spontan und in einer begrenzten Zeitspanne (z.B. in einer kurzen Pause oder während einer Busfahrt) ein Spiel spielen. Das „Schnelle Spiel“ hilft ihm auch, einen schnellen, ersten Eindruck vom Spiel zu erhalten.   **Vorbedingungen:** Der Spieler hat die Anwendung geöffnet.  **Nachbedingungen:** Die Spielstatistiken sind gespeichert und der Highscore wurde mit dem Punktestand des Spiels aktualisiert. Der Spieler ist wieder im Hauptmenü der Anwendung.  **Standardablauf:**   1. Der Spieler wählt im Hauptmenü „Schnelles Spiel“. 2. Innert 10s (oder weniger) lädt das Spiel. 3. Der Spieler befindet sich nun im regulären Spielbildschirm. Der (interne) Timer steht auf 60s. Es verbleiben 3 Versuche. 4. Im oberen Bildschirmabschnitt bewegt sich ein Güterzug mit Containern von links in das Spielgeschehen hinein. Ab diesem Moment beginnt der (interne) Timer herunter zu zählen. 5. Der Spieler bestimmt, wo er den vordersten Container des Güterzugs auf dem Containerschiff platzieren möchte. 6. Das Spiel zeigt eine Animation des Hafenkrans, welcher den Container auf der vom Spieler bestimmten Position ablegt. 7. Das Spiel berechnet die Gewichtsverteilung auf dem Containerschiff neu. 8. Die Schritte 5 bis 7 werden so lange wiederholt, bis der (interne) Timer 0 erreicht. Der Güterzug bringt dann keine neuen Container mehr. 9. Der Spielablauf ist zu diesem Zeitpunkt zu Ende. Das Spiel berechnet anhand der Gewichtsverteilung und Höhe der Containerstapel den Punktestand. 10. Das Spiel zeigt als Schlussanimation an, wie das Containerschiff den Hafen verlässt. Anschliessend wird der erreichte Punktestand angezeigt.   **Erweiterungen:**  8a. Der vorderste Container auf dem Güterzug erreicht den rechten Bildschirmrand.   1. Eine Animation wird abgespielt, die dem Spieler signalisiert, dass dieser Container verloren ist. 2. Es wird ein Versuch abgezogen.   2a. Handelt es sich um den letzten Versuch, gilt das Spiel als verloren (Game Over) und der Spieler erhält keine Punkte.  10a. Die Gewichtsbelastung auf dem Containerschiff ist zu ungleichmässig.  1a. Falls die Gewichtsbelastung am Bug bzw. Heck des Containerschiffs unverhältnismässig gross ist, wird eine Animation angezeigt, wie das Containerschiff nach vorne bzw. hinten kippt und sinkt.  1b. Falls die Gewichtsbelastung an einem inneren Abschnitt des Containerschiffs unverhältnismässig gross ist, wird eine Animation angezeigt, wie es an der überbelasteten Stelle zerbricht und anschliessend sinkt.  2. Das Spiel gilt als verloren (Game Over) und der Spieler erhält keine Punkte.  **Spezielle Anforderungen:**   * Die Ladezeit (siehe Schritt 2) beträgt maximal 10 Sekunden. * Die Sprache ist Englisch * Die Bedienung erfolgt über Touch-Eingaben und ist möglichst intuitiv.   **Liste der Technik- und Datenvariationen:** -  **Häufigkeit des Auftretens:** Erwartungsgemäss sehr häufig. Die Frequenz ist natürlich vom Verhalten des Spielers abhängig. Das „Schnelle Spiel“ dürfte hauptsächlich unter folgenden Bedingungen aufgerufen werden:   * Der Spieler hat nur kurz Zeit für ein Spiel und möchte möglichst schnell Einsteigen * Der Spieler hat kein Interesse am Karriere-Modus * Der Spieler hat den Karriere-Modus bereits durchgespielt.   **Verschiedenes:**  Abzuklären / zu definieren:   * Wird der Timer (die ablaufende Zeit) für den Spieler sichtbar dargestellt? * Kann das Containerschiff auch bereits während dem Spielablauf sinken oder zerbrechen, anstatt erst nach Ablauf der Spielzeit? Die Auswirkung auf das Gameplay wäre, dass der Spieler zu jedem Zeitpunkt im Spiel gezwungen wäre, effizient zu stapeln, anstatt erst am Ende einen effizienten Zustand zu erreichen. Welche Option ein besseres Spielerlebnis bietet, zeigt sich u.U. erst während der Entwicklung / dem Playtesting.   Anmerkungen:   * Hier wird mit 60 Sekunden Spielzeit für das „Schnelle Spiel“ gerechnet. Evtl. zeigt sich während der Entwicklung/dem Playtesting, dass eine andere Zeitspanne sinnvoller wäre. |