

Entwicklungsprojekt: Audit 4

Meike Jungilligens, Mauricio Köppen, Fabian Ngo

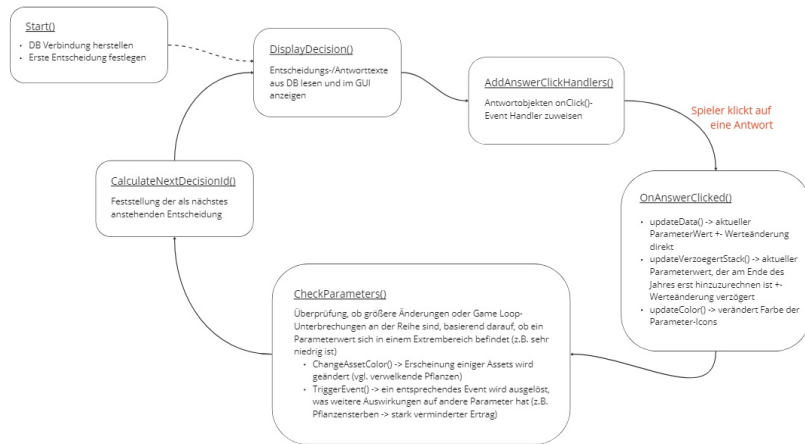


Gliederung

1. **Veränderungen im Code**
2. **Neuerungen in der Szene**
3. **Fazit und Reflektion**

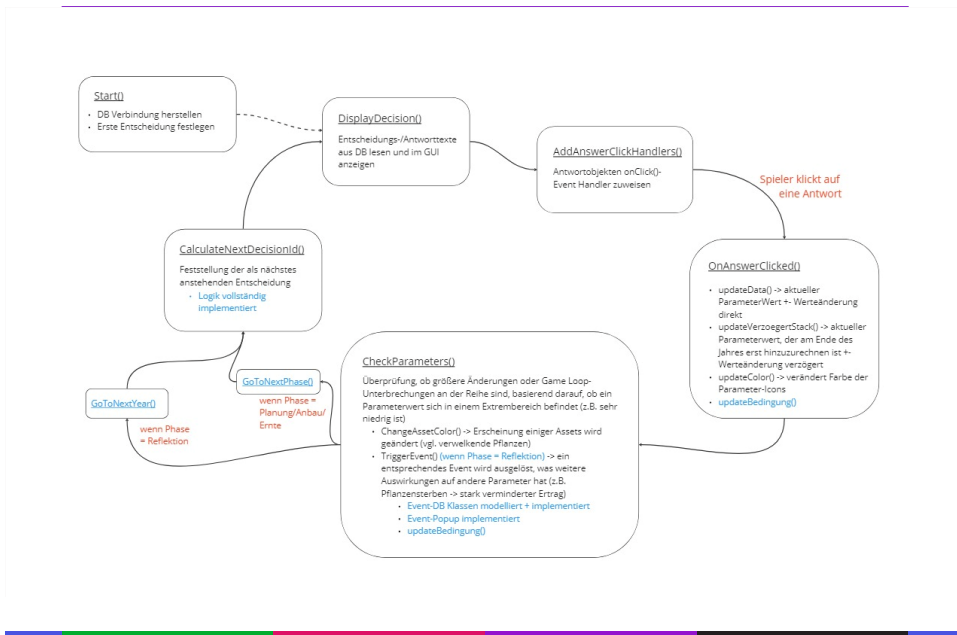


Veränderungen im Code: Main Game Loop alte Version



MGL Diagramm aus Audit 3 (hinzugefügt, um Veränderungen sichtbar zu machen)

Veränderungen im Code: Main Game Loop aktuelle Version



Dieses Modell zeigt zusammengefasst die Veränderungen in der Anwendungslogik seit Audit 3. In den folgenden Folien wird auf einige Änderungen noch genauer eingegangen. Zusammengefasst wurden Funktionen implementiert, die beim letzten Mal nur konzeptionell (oder vereinfacht) vorlagen (dazu gehören die DB-Logik der Events zum Jahresende und die Logik zur Auswahl der nächsten

Entscheidung sowie eines passenden Events am Jahresende), die Anpassung des UI (Event-Popup implementiert sowie Textänderung der Phase/des Jahres), und die Einführung von Bedingungen, um bestimmte Entscheidungen zu triggern.

Die Funktionen GoToNextPhase() und GoToNextYear() sind primär dazu da, den Text im UI, welcher dem Spieler die aktuelle Phase und das JAhr anzeigt, zu aktualisieren.

Gleichzeitig wird die Phase-Variable auch dafür genutzt, die aktuelle Phase zu erfahren.

So zum Beispiel in der

Entscheidungsgenerierung: In den Pool der möglichen Entscheidungen, die dem Spieler angezeigt werden können, sollen schließlich nur solche sein, die zur aktuellen Phase gehören. Dafür wird die UI-Variable

herangezogen. Genau so wird auch überprüft, wann die letzte Phase des Jahres (REFLEKTION) eingetreten ist, um die verzögerten Werte verrechnen und ein Event einblenden zu können.

Die Event-Tabelle in der Datenbank wurde ebenfalls implementiert (und an die hier getroffenen Änderungen angepasst) und besteht jetzt aus den Spalten event_id, event_text, affected_parameter_id, aenderung_direkt, bedingung_erfüllt, related_parameter_id

Erklärung zu den Spaltennamen:

Event_id/event_text: Selbsterklärend

Affected_parameter_id: Manche Events haben, genauso wie normale Entscheidungen, eine Auswirkung auf (einen) bestimmte(n) Parameter. Die ID dieses Parameters wird hier

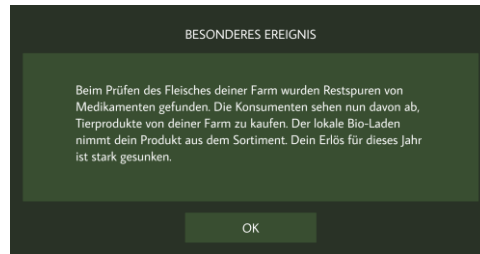
hinterlegt.

Aenderung_direkt: die Werteveränderung, die auf den eben genannten Parameter zukommt.

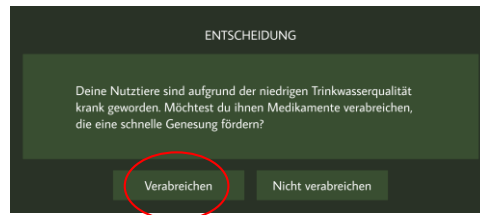
Bedingung_erfuellt: Wird in den Folgenden Folien erklärt

Related_parameter_id: Wie schon im letzten Audit definiert, wird ein Event bezogen auf einen Parameter getriggert, wenn dieser (z.B.) besonders niedrig ist. Um zu wissen, welche Events auf welchen Parameter bezogen (engl. "related") sind, wird hier die entsprechende Parameter-ID hinterlegt

Veränderungen in der Datenbankstruktur: Einführung von Bedingungen



Das Event setzt diese Entscheidung voraus:



In unserer Konzeption von den zu treffenden Entscheidungen waren wir nicht selten mit dem Fall konfrontiert, dass ein bestimmter Entscheidungsprompt oder ein bestimmtes Event erst Sinn macht, nachdem eine andere bestimmte Entscheidung getroffen wurde. Ein Beispiel dafür wären Events/Entscheidungen, die sich mit

direkten Konsequenzen anderer Entscheidungen befassen oder aufeinanderfolgende Entscheidungen, die erst in der Masse der Ausführung starke Konsequenzen mit sich tragen (wie zum Beispiel ein monokultureller Anbau über mehrere Jahre hinweg). Wie also halten wir für eine Entscheidung/ein Event fest, ob die ihr vorrausgesetzten Entscheidungen überhaupt schon getroffen wurden?

Veränderungen in der Datenbankstruktur: Einführung von Bedingungen

Antwortmöglichkeit					
<u>Antwort-ID</u>	...	Antworttext	...	eid_bedingung_erfuellt	evid_bedingung_erfuellt
01	...	"Verabreichen"	...	NULL	001
02	...	"Nicht verabreichen"	...	NULL	NULL

Event			
<u>event_id</u>	event_text	...	bedingung_erfuellt
001	Beim Prüfen des Fleisches deiner Farm wurden Restspuren von Medikamenten gefunden. Die Konsumenten...	...	0

bedingung_erfuellt
1

(evid = Event_ID; eid = Entscheidung_ID. Hat eine Antwort auf kein Event und keine Entscheidung einen "vorraussetzenden Einfluss", ist der Wert in der entsprechenden Spalte NULL)

Um dieses Problem ohne redundante If-Abfragen im Code lösen zu können, w

urde bei den DB-Tabellen für die Entscheidungen oder Events eine Spalte "bedingung_erfuellt" hinzugefügt, die den Wert 0 (false) oder 1 (true) beinhaltet. Bei der Auslese einer Entscheidung oder eines Events aus der DB wird dann geprüft, ob die Voraussetzungen für diese erfüllt wurden (also in der Spalte bedingung_erfuellt eine 1 steht), ansonsten wird eine andere ID generiert und zurückgegeben.

Einige Entscheidungen/Events haben keine besonderen Voraussetzungen, die erfüllt werden müssen, bei diesen ist der Wert in der Spalte standardmäßig auf 1. Bei allen anderen (mit Wert 0) kann der Wert durch das Auswählen einer bestimmt

en Antwort bei der vorausgesetzten Entscheidung auf

1 geändert werden. Im obigen Beispiel bewirkt also

das Auswählen der Antwort "Medikamente verabreichen", dass der Wert von

"bedingung_erfuellt"

des abgebildeten Events von 0 auf

1 geändert wird (Die ID des

Events, dessen bedingung_erfuellt-

Wert geändert wird, ist in der Antwort-

Tabelle in

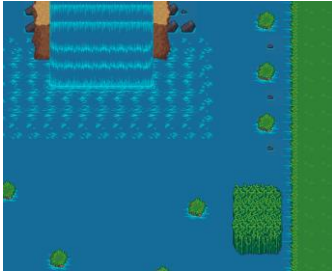
der Spalte "evid_bedingung_erfuellt"

hinterlegt). Dieses

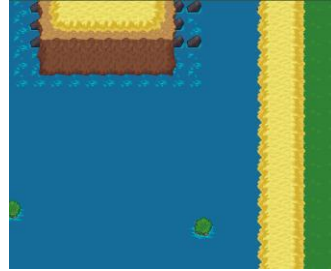
Event kann somit ausgelöst werden, da ja nun

die inhaltlichen Voraussetzungen dafür erfüllt sind.

Standard-Szene



Neue-Szene nach
bestimmter Bedingung



Anstatt das Sprite-Changer Skript zu verwenden haben wir uns entschieden ein neues Skript, nämlich das Change-Scene Skript zu verwenden. Es wurde sich dafür entschieden da das Sprite-Changer Skript gut für simple Veränderungen ist, aber bei komplexeren Veränderungen wie der Veränderung einer ganzen Umgebung an seine Grenzen kommt. Deswegen

nutzen wir Change-Scene Skript was nun die ganze Szene ändert, das hat den Vorteil das man so viel verändern kann wie man möchte ohne jedes einzelne Tile in das Skript hinzuzufügen. So musste zwar für die jeweilige Veränderung eine neue Szene angelegt werden aber es war dennoch weniger Arbeit als jede einzelne Veränderung in das vorherige Skript zu schreiben. Das Skript wechselt je nach Erfüllung der Bedingungen zur jeweiligen Szene.

Reflektion

1. Abstraktion gegenüber Realismus
2. Forschungsbeschränkung
3. Zielgruppenansprache
4. Einflussnahme auf die Debatte
5. Einarbeitung in die Game-Engine
6. Weiterentwicklungsmöglichkeiten

1. Abstraktion gegenüber Realismus

Der Ansatz, die komplexe Realität der Landwirtschaft auf vier vereinfachte Parameter zu reduzieren, ermöglicht es, das Spiel für Spieler zugänglicher zu machen. Allerdings birgt dies auch die Gefahr, dass wichtige Aspekte der Realität verloren gehen oder verzerrt werden.

2. Forschungsbeschränkung

Es wurde mangels Ressourcenknappheit keine umfangreiche Recherche in dem Ausmaß aufgenommen, die der Komplexität der Landwirtschafts-Domäne für eine realitätsgetreue Simulation entsprechend nötig wäre.

3. Zielgruppenansprache
Spieler ohne Vorkenntnisse sollten einen leichteren Einstieg durch die einfachen Parameter haben. Die Spielmotivation liegt in einem Wiederspielwert begründet, die durch Fehlentscheidungen des Spielers hervorgerufen werden. Durch das Tragen der Konsequenzen seiner Entscheidungen, ist der Spieler gezwungen damit umzugehen und ist motiviert eventuell bei nächsten

Spieldurchlauf erfahrungsbasiert
bessere Entscheidungen zu treffen
und nebenbei über die Praktiken in
der Landwirtschaft zu lernen.

Ob dies wirklich der Fall ist, wird
sich beim Testen des Spiels mit
einer fundierten Datenbank zeigen
müssen.

4. Einflussnahme auf die Debatte
Das Ziel mit dem Spiel zukünftig
einen konstruktiven Beitrag für die
Debatte über eine nachhaltige
Landwirtschaft zu leisten, ist von
der praktischen Umsetzung mit
einer fundierten Datenbank, und der
Anzahl aktiver Spieler abhängig.
Zum aktuellen Zeitpunkt kann dies
nicht beantwortet werden.

5. Einarbeitung in die Game-Engine
Die Einarbeitung des Teams in Unity 3D

gelang planmäßig und weitgehend
Problemlos. Die Kompetenzen, um einen
funktionalen Prototypen zu erstellen,
wurden angeeignet.

- Datenbankanbindung

- UI-Entwicklung

6. -Asset-Management

- Skripting mit Unity-GameObjects

7. Weiterentwicklungsmöglichkeiten

- Integration zusätzlicher Parameter

- Zusammenarbeit mit Experten

- Nutzung externer Datenquellen


- Eigene intensive Recherche

Fazit

Es wurde ein funktionaler Prototyp umgesetzt, der die Kernfunktionalitäten eines interaktiven entscheidungsbasierten Landwirtschaftssimulators abbildet.

Im Laufe des Entwicklungsprozesses wurde die Komplexität der Domäne der Landwirtschaft unterschätzt, sodass ein ursprüngliches Ziel, die Umsetzung einer Datenbank mit fundierten Daten, mit den Ressourcen des Teams nicht umsetzbar war.

Dennoch bietet das Projekt eine solide Grundlage für weiterführende Entwicklungen und Verbesserungen. Durch die Integration zusätzlicher Parameter, eine engere Zusammenarbeit mit Experten und die Nutzung externer Datenquellen könnte das Spiel seine Wirksamkeit steigern.



Das Projekt zielt darauf ab, die komplexe Realität und Herausforderungen der Landwirtschaft durch ein interaktives entscheidungsbasiertes Spiel zugänglicher zu machen. Durch die Entwicklung eines funktionalen Prototyps, worin Spieler die Rolle eines Kleinbauerns einnehmen, Verantwortung für ihren simulierten Hof übernehmen und Entscheidungen treffen, sollen sie für die Herausforderungen der

Landwirtschaft sensibilisiert werden.

Im Laufe des Projektes wurde festgestellt, dass die Vereinfachung der Realität auf vier Hauptparameter zwar den Zugang zum Thema erleichtert, aber auch die Gefahr birgt, wichtige Aspekte zu vernachlässigen. Die Beschränkungen in Bezug auf Ressourcen für umfassende Forschung und Datenbanken könnten die Authentizität und Relevanz des Spiels beeinträchtigen. Es ist wichtig zu überprüfen, ob das Spiel tatsächlich effektiv ist, um Spieler dazu zu bringen, sich mit den komplexen Herausforderungen der Landwirtschaft auseinanderzusetzen. Dennoch bietet das Projekt eine solide Grundlage für weiterführende Entwicklungen und Verbesserungen. Durch die Integration zusätzlicher Parameter, eine engere

Zusammenarbeit mit Experten und die Nutzung externer Datenquellen könnte das Spiel seine Wirksamkeit steigern und einen noch größeren Beitrag zur Sensibilisierung für die Herausforderungen der Landwirtschaft zu leisten. Insgesamt ist das Projekt ein Schritt in Richtung einer breiteren Diskussion und Sensibilisierung für die Herausforderungen und Chancen in der Landwirtschaft.