

Herausforderung Spieleumbau

Ausgangssituation:

Ein beliebtes und bewährtes Spiel einer Website soll neu aufgelegt werden, damit es auch auf Tablets und Smartphones gespielt werden kann. Momentan ist das Spiel aus einem oder mehreren Gründen mobil nicht verfügbar, da:

- Anwendung wurde mit Flash erstellt, keine Ausspielung auf Tablet/Smartphone möglich.
- Spiel ist optimiert für Desktop, verfügt über feste Größe, ist nicht responsive.
- Steuerung des Spiels erfolgt über Tastatur oder Mausektionen, die nicht auf Touchdevices funktionieren, z.B. Mausverfolger oder Hover

Analyse:

Was kann von der alten Anwendung übernommen werden?

- Spielprinzip :)
- Illustrationen (Animationen)
- Sounds/Musik
- Einzelne Funktionen und Logik des Programmcodes (ActionScript -> JavaScript)

Zu klären wäre weiterhin:

- Liegt die FLA-Datei (Flash-Quellcode) vor?
- Auch wenn Arbeits-Dateien vorliegen, heißt das nicht automatisch, dass diese für eine Neuauflage des Spiels benutzt werden dürfen. Die Nutzungsrechte für Grafik, Illustrationen, Fotos und Sounds sind zu klären.

Produktion:

Was muss neu gemacht werden?

- Konzeption für „Responsive“:
Mobile First: Das Spiels wird zuerst für die Smartphone-Version konzipiert, danach wird die Konzeption auf größere Auflösungen ausgeweitet. Dabei gilt: Identischen Funktionsumfang auf allen Devices ermöglichen, die Darstellung/Anordnung darf abweichen.
- Anpassung des Grafik Design/Raster/Layout z.B. Buttongröße, Menüs, Definition von Breakpoints
- Evtl. Steuerungsmechanik z.B. Drag'n Drop, Virtueller Joystick, Swipe usw.
- „Faire Wettbewerbsbedingungen“ auf allen Devices schaffen
- Programmierung (inkl. Integration von Animationen und Sounds)
- Crossbrowser- und Device-Testing

Gut zu wissen! Was muss beachtet werden?

- Anforderungen an ein Spiel sind sehr unterschiedlich, daher gibt es hier keine Allroundlösung und daher auch selten ein Framework, welches benutzt werden kann: eigentlich sprechen wir hier immer über Handarbeit/manuelle Programmierung.
- Animationen z.B. Intros, die synchron zum Ton laufen sind schwer realisierbar: evtl. besser auf Video ausweichen oder anders lösen.
- Nicht alle Animationen können ebenso elegant umgesetzt werden wie mit Flash. Es ist darauf zu achten, dass kein Qualitätsverlust durch „lieblos erscheinende“ Animationen entsteht.
- Es lässt sich nicht alles 1:1 nachbauen. Es lässt sich aber sehr Vieles sehr gut nachbauen. Es gibt unterschiedlichste Möglichkeiten, die eine qualitativ hochwertige Produktion erlauben.
- Die Dateigröße sollte nicht erheblich größer werden als bei der Flashversion, besonders wichtig für Zugang ohne WLAN z.B. PNG komprimieren (PNG Crusher/TinyPNG), SVG benutzen, HTML5 Picture-Tag
- Ggfls. Texte für Mobile kürzen

Budget

Faustformel:

- Der Umbau eines Spiels benötigt ungefähr das gleiche Investitionsvolumen wie die damalige Flash-Produktion, wenn Teile des Spiels wiederverwendet werden können.
- Die Neuproduktion eines Spiels in HTML5 ist zumeist teurer als die Umsetzung in Flash

Chancen

Zusätzlich zu strukturellen Anpassungen der grafischen Oberfläche (z.B. Anordnung der Elemente) kann das gesamte Grafik Design mit überschaubarem Aufwand zeitgemäß aufgefrischt werden. Somit erhält das Spiel einen weitaus größeren Mehrwert, als bei einem rein „technischen“ Relaunch.

Einbindung

Im Wesentlichen kann zwischen zwei Einbindungsmöglichkeiten unterschieden werden:

1. Das Spiel wird als Modul auf einer Website eingebunden.
2. Das Spiel stellt eine eigene Microsite dar.

Bei der Einbindung eines Spiels als Modul sind folgende Punkte zu beachten:

- Ist die Website, in die das Spiel integriert werden soll selbst responsive?
- Sollen neben dem Spiel noch zusätzliche Elemente sichtbar sein (Hauptmenü, Header, Footer usw.) oder werden diese ausgeblendet?
- Das Scrollen der Website sollte ggfls. deaktiviert werden, sobald das Spiel aktiv ist

- Neupositionierung des Spiels in den Viewport, sobald das Device zwischen Portrait- und Landscape-Modus wechselt.

Wird ein Spiel als eigenständige Microsite eingebunden, sind obige Punkte nebensächlich. Microsite-Spiele sollten unbedingt mit folgende Angaben ausgestattet werden:

- Impressum
- Kontakt
- Angaben zum Datenschutz

Video

Wir unterscheiden im Wesentlichen zwischen zwei unterschiedlichen Arten der Videointegration in eine Website.

1. Das Video wird selbst gehostet
2. Das Video wird über ein Videoportal ausgeliefert

Pro Selbst gehosted

- Kontrolle über die Daten. Auf Video-Plattformen kann das Video gebanned/gesperrt werden
- Auf Videoplattformen verleiten Teaser und Links am Ende des Videos den User, die Website zu verlassen. Diese Links können durch Code (auf der eigenen Seite) unterdrückt werden, aber die APIs ändern sich häufig und müssen daher überwacht werden.
- Mehr direkten Traffic. Wird das Video empfohlen, gelangen Benutzer nur auf Eure Seite.
- Kontrolle über das Player-Design. (Muss erstellt und programmiert werden)
- Keine Werbung!
- Kostenpflichtiger Traffic

Pro Video-Plattform z.B. Youtube oder Vimeo

- Automatische Konvertierung der Videos
- Bereitstellung verschiedener Auflösungen, Datenraten und Fallbacks für ältere Systeme
- Gutes SEO z.B. Youtube-Videos werden bei Google-Suchergebnissen gelistet
- Social Traffic: User, die ähnliche Videos ansehen/suchen, „stolpern“ über Euer Video
- Einfache Möglichkeit Video mit Untertiteln zu versehen
- Kostenloser Traffic

Wenn das Video selbst gehostet wird. Dann wird es meist über das HTML5 Video –Tag eingebunden (Das Video wird dann nicht gestreamt, sondern progressiv geladen)

Die unterschiedlichen Browser und Devices verlangen/bevorzugen unterschiedliche Videoformate.

Standards sind:

```
<video>
<source src="video.m4v" type="video/mp4" />
<source src="video.webm" type="video/webm" />
<source src="video.ogv" type="video/ogg" />
</video>
```

Der jeweilige Browser entscheidet, welches Video er lädt und abspielt.
Mit diesen drei Formaten ist auf jeden Fall ein Video sichtbar :)

Jeder Browser bringt seinen eigenen Videoplayer mit. Die Controls/Steuerelemente sind nicht stylebar. Das heißt, in jedem Browser sieht der Player, welcher das Video umgibt etwas anders aus. Es gibt die Möglichkeit, per Javascript selber Controls zu programmieren und diese zu stylen. Dies ist allerdings auch mit Arbeitszeit verbunden. iPad und iPhone nutzen den nativen Player, um Videos abzuspielen.

Nicht jeder Browser zeigt das Bild des ersten Frames des Videos an. Daher wird empfohlen ein Videoposter zu integrieren.

Einige Browser zeigen einen großen Playbutton über dem Video/Videoposter.

Einige Devices unterbinden das automatische Abspielen von Videos z.B. iPhone. Die Videos müssen aktiv über eine Buttonaktion gestartet werden.

Empfehlung:

Anders als vor noch 5 Jahren ist die Akzeptanz der browsereigenen Controls gestiegen. Diese werden von den Benutzern nicht mehr als Fremdkörper angesehen.

Tipp: Geld sparen! Auf selbstgestaltete Steuerungselemente verzichten und akzeptieren, dass es in den Browsern unterschiedlich aussieht.

FLV Flash-Video

Flash-Videos (FLV) können leicht in oben beschriebene Videoformate konvertiert werden. z.B. Adobe Media Encoder oder EasyHTML5Video. Es gibt auch Freeware.

Audio

Ähnlich wie bei den Videos verhält es sich auch mit den Audiodateien:

Die Audio-Standards lauten:

```
<audio>  
<source src="audio.ogg" type="audio/ogg">  
<source src="audio.mp3" type="audio/mpeg">  
<source src="audio.wav" type="audio/wav"> Achtung Dateigröße!  
</audio>
```

Ebenso wie die Controls des Videoplayers sind die Controls des browsereigenen Audioplayers nicht stylebar.