

Politechnika Łódzka

Wydział Elektrotechniki, Elektroniki, Informatyki i Automatyki

Katedra Mikroelektroniki i Technik Informatycznych

PRACA DYPLOMOWA INŻYNIERSKA

Aplikacja webowa wspomagająca proces zapamiętywania

Autor: Emil Jasiński

Numer albumu: 164357

Kierujący pracą:
dr inż. Wojciech Tylman

Łódź, luty, 2014

Spis treści

Wstęp	5
Rozdział 1. Cel i zakres pracy	7
Rozdział 2. Mobilne aplikacje webowe	9
2.1. Historia specyfikacji HTML5	9
2.2. Różnice w stosunku do HTML4	9
2.3. Visual Studio i Asp.Net jako środowisko pracy	13
Rozdział 3. Algorytm SuperMemo	15
3.1. Krzywa zapominania	15
3.2. Opis	15
3.3. Implementacja	15
Bibliografia	17
Spis rysunków	19
Spis tabel	21
Załączniki	23
Streszczenie (Summary)	25

Wstęp

W obecnych czasach z każdej strony do naszego mózgu dociera masa zbędnych informacji. Część bez naszej woli jak na przykład świecący billboard reklamowy, na który mimowolnie spoglądamy jadąc samochodem, a na część decydujemy się sami gdy naszą pierwszą decyzją po przebudzeniu jest włączenie telewizora lub komputera.

Coraz więcej informacji jest nam również potrzebne w życiu zawodowym. Wpis o znajomości jednego języka obcego w CV nie wyróżnia nas już na tle innych ubiegających się o pracę. Niektóre profesje, jak na przykład zawód programisty wymagają nieustannego podnoszenia swoich kwalifikacji.

Taki nadmiar informacji, którego nie jesteśmy w stanie przetworzyć nazywa się przeciążeniem informacyjnym i może skutkować pogorszeniem się naszego samopoczucia i uczuciem przytłoczenia. Rozwiązaniem na ten rosnący problem społeczny jest wygospodarowanie każdego dnia czasu na spacer czy słuchanie muzyki, która nas odpręża. Ulgę naszemu organizmowi może też przynieść skrócenie czasu poświęconego na naukę. Okazuje się, że istnieje wiele sposobów na ułatwienie zdobywania nowej wiedzy są to między innymi:

- umiejętność szybkiego czytania
- mnemotechnika
- metoda SuperMemo

Ostatnia z wymienionych metod oraz aplikacja implementująca algorytm o tej samej nazwie jest przedmiotem tej pracy.

Projekt aplikacji przedstawionej w rozdziale "Projekt aplikacji wspomagającej zapamiętywanie" powstał by w jak największym stopniu spełnić wymagania potencjalnych użytkowników. Założono, że użytkownik może chcieć korzystać z aplikacji na swoim smartphonie, tablecie czy komputerze, bez dostępu do internetu gdy nie jest to niezbędne. Przy dodatkowym założeniu, że projekt nie wymaga dużej mocy obliczeniowej czy dostępu do natywnych elementów systemu operacyjnego, zdecydowano się na wykorzystanie zbioru standardów oferowanych przez HTML w wersji piątej.

Rozdział 1

Cel i zakres pracy

Celem pracy jest zaprojektowanie aplikacji, której pierwsza część pozwoli użytkownikowi utworzyć bazę wiedzy składającą się z pytań i odpowiedzi. Druga część służy do prezentowania pytania. Użytkownik ma na tym etapie zastanowić się nad odpowiedzią. Gdy zna już odpowiedź lub uznaje, że przypomnienie sobie odpowiedzi trwa zbyt długo, przechodzi do części prezentacji odpowiedzi z formularzem, w którym sami oceniamy swój poziom zapamiętania tego pytania. Jest to sposób przyjęty w konkurencyjnych aplikacjach wykorzystujących algorytm SuperMemo.

Zakres pracy przedstawia się następująco:

- Test przeglądarek internetowych pod kątem obsługi HTML5
- Opis użytych technologii ze standardu HTML5
- Przegląd cech Visual Studio i Asp.Net, które ułatwiają pisanie nowoczesnych aplikacji webowych
- Historia algorytmu SuperMemo
- Implementacja algorytmu SuperMemo w wersji drugiej
- Zaprojektowanie aplikacji wykorzystującej algorytm
- Testy aplikacji

Rozdział 2

Mobilne aplikacje webowe

2.1. Historia specyfikacji HTML5

Html (ang. HyperText Markup Language) jest językiem znaczników używanym do stworzenia struktury i wyglądu strony WWW (ang. World Wide Web). Pierwsza specyfikacja została upubliczniona w roku 1991 przez fizyka Tima Bernersa-Lee, który wcześniej stworzył prototyp hipertekstowego systemu informatycznego, pracując dla ośrodka CERN. Ideą, która zdecydowała o sukcesie takiego systemu prezentowania treści w sieci Internet były odnośniki. W ten sposób na jednej stronie można było zgrupować hiperłącza do wyników badań ośrodków naukowych z wielu miejsc na świecie.

Ostania wersja specyfikacji to HTML 4.01, która została udostępniona przez organizację W3C (ang. World Wide Web Consortium) w roku 1999, jednak już w 1998 W3C podjęło decyzję o zatrzymaniu rozwoju specyfikacji HTML i rozpoczęciu prac nad bazującą na XML alternatywą nazwaną XHTML. Kontrowersje wzbudziła specyfikacja XHTML 2.0, która nie była kompatybilna z wcześniejszymi wersjami.

W roku 2004, pracownicy firm Apple, Mozilla Foundation i Opera założyli grupę WHATWG (ang. Web Hypertext Application Technology Working Group). Było to spowodowane obawami przed kierunkiem rozwoju jaki przyjęło W3C - brak rozwoju HTML i brak odpowiedzi na potrzeby twórców witryn i przeglądarek internetowych. Grupa zaczęła pracować nad standardem HTML w wersji piątej.

W roku 2006 W3C zgodziło się użyć propozycji grupy WHATWG. Grupa W3C powołana do rozwoju XHTML 2 zaprzestała prac w 2009. W3C postanowiło przyspieszyć rozwój HTML 5 inwestując w pracę nad nim swoje zasoby.

W roku 2014 HTML 5 stanie się rekomendacją W3C. Na rok 2016 planowane jest opublikowanie HTML 5.1.

2.2. Różnice w stosunku do HTML4

HTML 5 jest stale rozwijającym się standardem. Projektując stronę używając HTML 4 nie musimy martwić się o aktualizowanie strony, o ile oczywiście nie chcemy

dodać nowych funkcjonalności. Strona zbudowana w oparciu o niego, która przeszła proces walidacji powinna wyświetlać się identycznie na wszystkich przeglądarkach niezależnie od daty powstania. Jeśli przejdziemy na HTML 5 może dojść do sytuacji, że wykorzystany przez nas kilka miesięcy wcześniej element lub atrybut zmienił się. Dotyczy to szczególnie elementów wprowadzonych w HTML 5 (np. znaczniki audio, video). W związku z tym uproszczeniu mogła ulec deklaracja typu dokumentu.

Zamiast

```
1 <!DOCTYPE HTML PUBLIC "-//W3C//DTD_HTML
2 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">
```

używamy:

```
1 <!DOCTYPE html>
```

Nowe znaczniki to między innymi:

canvas używany do rysowania grafiki przez skrypty pisane w języku java script

audio definiuje plik dźwiękowy jako treść strony

video definiuje plik wideo jako treść strony

article definiuje artykuł

nav definiuje sekcję, która zawiera tylko linki

address definiuje sekcję zawierającą informacje kontaktowe

main definiuje główną, najbardziej znaczącą część strony

dialog definiuje okno dialogowe

section definiuje sekcje dokumentu

footer definiuje stopkę dla całego dokumentu lub sekcji

header definiuje nagłówek dla całego dokumentu lub sekcji

mark definiuje zaznaczony tekst

progress definiuje postęp zadania

time definiuje datę/czas

API (ang. Application Programming Interface) HTML 5 udostępnia programiście wiele nowości (rys. 2.1).

Ciasteczka (ang. cookies) to porcja danych wysyłana z przeglądarki internetowej i przechowywana jako mały plik tekstowy na dysku twardym użytkownika strony. Wykorzystywane są do przechowywania informacji o aktywności użytkownika (np. czy użytkownik jest zalogowany).

Zalety:

— nie wymagają zasobów serwera

	MAC						WIN									
																
	CHROME	FIREFOX	OPERA	SAFARI			CHROME	FIREFOX	OPERA							
	25	20	12	15	5.1	6	25	15	12	6	7	8	9	10		
Local Storage	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗	✓	✓	✓	95%	
Session Storage	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗	✓	✓	✓	96%	
Post Message	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗	✓	✓	✓	97%	
Offline Applications	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗	✗	✗	✓	85%	
Workers	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗	✗	✗	✓	82%	
Query Selector	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗	✓	✓	✓	95%	
Indexed Database	✓	✓	✗	✓	✗	✗	✓	✓	✗	✗	✗	✗	✗	✓	25%	
Drag and Drop	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	96%	
Hash Change (Event)	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗	✓	✓	✓	94%	
History Management	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗	✗	✗	✓	70%	
WebSockets	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗	✗	✗	✗	✓	56%	
GeoLocation	✓	✓	✓	✗	✓	✓	✓	✓	✓	✗	✗	✗	✓	✓	89%	
Touch	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	6%	
File API	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗	✗	✗	✓	29%	
Meter element	✓	✓	✓	✓	✓	✓	✓	✗	✓	✗	✗	✗	✗	✗	22%	
Progress element	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗	✗	✗	✓	28%	

Rysunek 2.1: Obsługa poszczególnych technologii HTML5 na najpopularniejszych przeglądarkach

Źródło: fmbip.com/litmus/

— są łatwe w implementacji

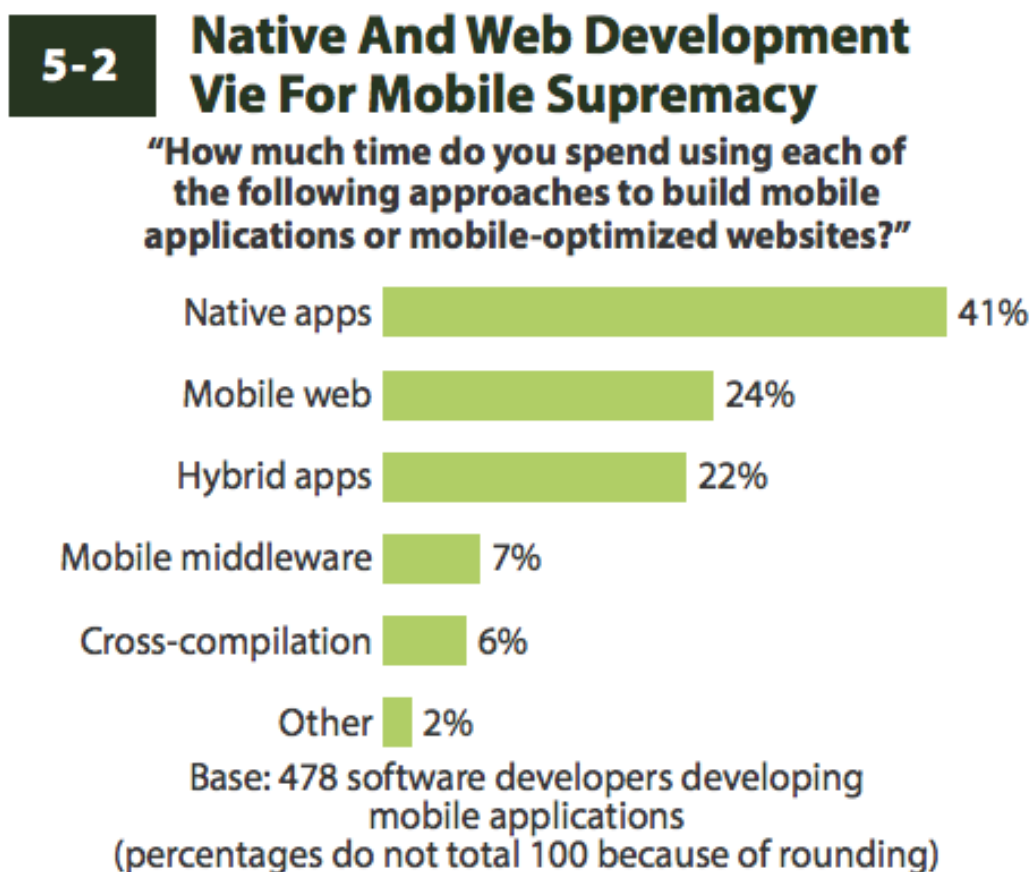
Wady:

- są każdorazowo wysyłane z żądaniem i odpowiedzią serwera
- mogą być wyłączone przez użytkownika przeglądarki
- problemy związane z bezpieczeństwem
- mała pojemność

Nowa specyfikacja proponuje alternatywę w postaci WebStorage oraz IndexedDB. WebStorage dzieli się na local storage i session storage. Dane zapisane w local storage nie wygasają. Dane w session storage wygasają po zamknięciu przeglądarki lub okna oraz są oddzielne dla każdej z instancji aplikacji.

IndexedDB jest bardziej zaawansowanym rozwiązaniem. Jest to interfejs lokalnej bazy danych, w której przechowywane mogą być zarówno wartości proste jak i skomplikowane, hierarchiczne obiekty. Rozwiązanie zostało zaproponowane przez firmę Oracle w roku 2009.

Zupełną nowością jest natomiast możliwość pisania aplikacji działających w trybie offline. AppCache pozwala twórcom witryn zdefiniować, które pliki powinny zostać zapisane na dysku użytkownika, a następnie być dostępne w trybie offline. Aplikacja napisana w ten sposób załaduje się i będzie działać poprawnie nawet gdy użytkownik jest poza zasięgiem sieci pod warunkiem, że już kiedyś odwiedził daną witrynę. Można



Source: Forrsights Developer Survey, Q1 2013

Rysunek 2.2: Popularność sposobów budowy aplikacji mobilnych

Źródło: www.forrester.com

więc przyrównać pierwsze władowanie strony w przeglądarce do instalacji natywnych aplikacji (np. w sklepie Google Play w systemie Android). By następnie zaktualizować zapisane na dysku pliki należy zmienić manifest (plik tekstowy, który zawiera między innymi spis plików).

Zalety:

- użytkownicy mogą nawigować po stronie bez dostępu do internetu
- zysk na szybkości - zasoby naszej aplikacji ładowane są bezpośrednio z dysku

Powyższe cechy powodują, że warto przed rozpoczęciem pisania aplikacji na urządzenia mobilne przemyśleć zalety i wady wybrania podejścia - zaprojektuj raz, uruchamiaj wszędzie, oferowanego przez HTML5 i język javascript. Zainteresowanie programistów tą technologią możemy odczytać z rys. 2.2.

2.3. Visual Studio i Asp.Net jako środowisko pracy

Wybierając edytor lub zintegrowane środowisko programistyczne do budowy aplikacji webowej warto sprawdzić czy posiada on zestaw pluginów Zen Coding. Znacząco przyspiesza to pisanie kodu html, a dzięki temu, że silnik pluginów jest niezależny od edytora powinniśmy znaleźć dodatek do każdego popularnego IDE (ang. Integrated Development Environment). Użycie tego dodatku wiąże się z wpisaniem w edytor odpowiedniej komendy i naciśnięciu klawisza TAB.

Przykład działania:

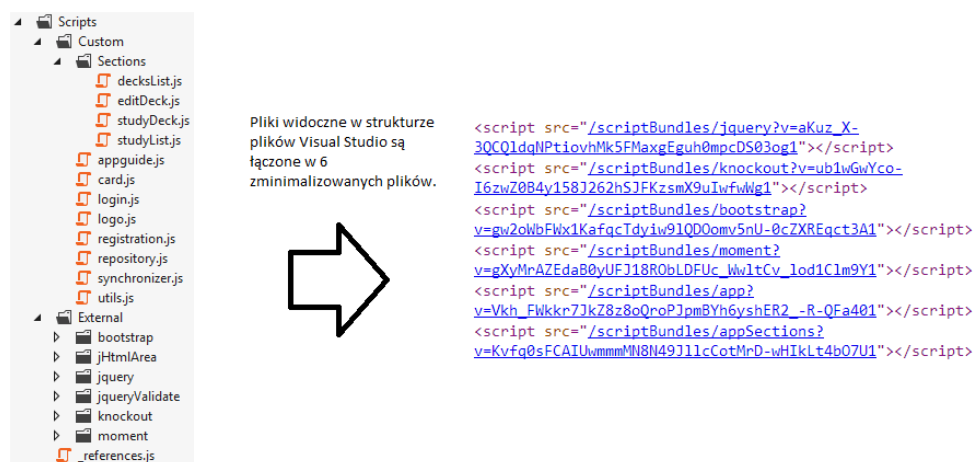
```
1 html:5>div#page>div.logo+nav>ul#navigation>li*5>a
```

Rezultat:

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <title></title>
5     <meta http-equiv="content-type" content="text/html; charset=UTF-8" />
6 </head>
7 <body>
8     <div id="page">
9         <div class="logo"></div>
10        <nav>
11            <ul id="navigation">
12                <li><a href=""></a></li>
13                <li><a href=""></a></li>
14                <li><a href=""></a></li>
15                <li><a href=""></a></li>
16                <li><a href=""></a></li>
17            </ul>
18        </nav>
19    </div>
20 </body>
21 </html>
```

Odpowiedni dodatek do Visual Studio można pobrać przez wbudowane narzędzie do pobierania dodatków jak i bibliotek programistycznych - NuGet. Natomiast funkcjonalnością, której nie możemy odtworzyć w innych IDE jest IntelliSense. Dzięki niej podczas wpisywania kodu w języku C# czy java script widzimy podpowiedzi odnośnie nazwy funkcji czy opis parametrów jak dana metoda przyjmuje. W przypadku tego drugiego języka podpowiedzi w innych IDE są rzadkością.[1]

W nowoczesnych aplikacjach na znaczeniu zyskała warstwa kliencka, a cały wygląd powinien być zdefiniowany poprzez kaskadowe arkusze stylów. Zasadą, która pozwala na oddzielenie kodu html, css i js jest tworzenie dla nich osobnych plików.



Rysunek 2.3: Widok plików w IDE oraz kod html połączonych plików

Źródło: Opracowanie własne

Powinny być ułożone w logiczny sposób np. każda klasa napisana w języku java script w osobnym pliku. Wiąże się z tym zwiększenie liczby żądań do serwera. Istnieje cecha Asp.Net dzięki której możemy zachować logiczną strukturę naszych plików warstwy klienckiej i jednocześnie zwiększyć optymalizację naszej witryny. Bundles, bo tak nazywa się ta funkcjonalność, pozwala nam utworzyć grupę plików, która zostanie połączona w jeden plik o podanej przez nas nazwie. Dodatkowo plik wynikowy zostanie zminimalizowany, co może znacząco obniżyć jego rozmiar. Minimalizację plików może wyłączyć w trybie DEBUG.[2]

Przeglądarki internetowe posiadają pamięć podręczną tzw. cache, który przechowuje pliki z wcześniej odwiedzanej strony. W ten sposób łącze internetowe klienta i zasoby serwerowe są odciążone, a strona ładuje się szybciej. Jest jednak istotna wada tego rozwiązania - klient może posiadać nieaktualny plik. Bundles oferują rozwiązanie dzięki dołączaniu do adresu z plikiem argument tzw. Content Hash. Jest on generowany na podstawie treści pliku. Gdy treść ulegnie zmianie zmieni się też hash. W ten sposób klient pobierze zawsze najnowszy plik. Jest to ważne gdy w naszych plikach znajduje się rozwiązanie krytycznego błędu naszej aplikacji.[2]

Powyższe cechy sprawiają, że Visual Studio i Asp.Net jest dobrym wyborem jeśli szukamy środowiska do pisania nowoczesnych aplikacji webowych.

Rozdział 3

Algorytm SuperMemo

3.1. Krzywa zapominania

Inaczej zwana krzywą Ebbinghausa, od nazwiska Hermanna Ebbinghausa. Ten niemiecki psycholog zaproponował w swojej pracy "O pamięci" zależność ilości pamiętanego materiału od czasu. Typowy wykres krzywej zapominania pokazuje, że ludzie zapominają połowę nowo nabytej wiedzy w ciągu kilku dni lub tygodni jeśli materiał nie był regularnie powtarzany. W 1885 Ebbinghaus wysnuł teorię o wykładniczej naturze zapominania i opisał ją wzorem:

$$R = e^{\frac{-t}{s}},$$

gdzie R - zachowanie wiedzy, S - względna siła pamięci, t - czas

Ebbinghaus uczył się sylab pozbawionych znaczenia poprzez powtarzanie po różnych okresach czasu. Ustalił w ten sposób, że spadek zapamiętanych informacji staje się coraz mniejszy. Krzywa zapominania stała się fundamentem do dalszych badań i w rezultacie opracowanie algorytmu SuperMemo.

3.2. Opis

Dr Piotr Woźniak prowadzi badania nad teorią metod nauczania, które zaowocowały powstaniem algorytmu SuperMemo i programu o tej samej nazwie. Jest on też osobą odpowiedzialną za komercyjne wykorzystanie tej metody. Pierwsza wersja programu powstała w 1987 roku. Napisana została w środowisku Turbo Pascal 3.0, IBM PC i powstała by zaprezentować metodę na dwa podstawowe sposoby:

- zastosowanie procedur optymalizacji na najmniejszych możliwych przedmiotach
- zróżnicowanie elementów na podstawie poziomu trudności

3.3. Implementacja

Bibliografia

- [1] Bruce Johnson, *Professional Visual Studio 2012* – Wrox, 2012
- [2] Adam Freeman, *Pro ASP.NET MVC 4* – Apress, 2012, rozdział 24
- [3] Piotr A. Wozniak, *Optimization of learning* – Master's Thesis
- [4] <http://www.supermemo.com/english/history.htm>
- [5] <http://zajackowski.org/2008/11/01/powtarzaj-i-odtwarzaj/>

Spis rysunków

2.1. Obsługa poszczególnych technologii HTML5 na najpopularniejszych przeglądarkach . . .	11
2.2. Popularność sposobów budowy aplikacji mobilnych	12
2.3. Widok plików w IDE oraz kod html połączonych plików	14

Spis tabel

Załączniki

1. Płyta CD z niniejszą pracą w wersji elektronicznej.
2. Oświadczenie o oryginalności pracy i możliwości jej wykorzystania.

Streszczenie (Summary)

Aplikacja webowa wspomagająca proces zapamiętywania

Nie ma zatem takiego człowieka, który kocha cierpienie samo w sobie, kto by do niego dążył lub chciał go doświadczyć, tylko dlatego, że jest to cierpienie, a dlatego, że czasami zdarzają się takie okoliczności, w których to cierpienie może doprowadzić go do jakiejś wielkiej przyjemności. Dając przykład banalny: któż z nas kiedyś nie podejmował się trudnego wysiłku fizycznego mając na względzie uzyskanie z tego korzyści? Kto ma jakiegolwiek prawo obwiniać człowieka, który wybiera przyjemność nie wiążącą się z przykrymi konsekwencjami, albo tego, kto unika takiego cierpienia, które nie prowadzi do przyjemności? Jednocześnie potępiamy ze słusznym oburzeniem i czujemy niechęć do ludzi, którzy są tak owładnięci urokami nietrwałej przyjemności, tak zaślepieni jej pragnieniem, że nie dostrzegają, iż następstwem ich postępowania będą z pewnością cierpienie i trudności.

TYTUŁ ANGIELSKI

Nor again is there anyone who loves or pursues or desires to obtain pain of itself, because it is pain, but because occasionally circumstances occur in which toil and pain can procure him some great pleasure. To take a trivial example, which of us ever undertakes laborious physical exercise, except to obtain some advantage from it? But who has any right to find fault with a man who chooses to enjoy a pleasure that has no annoying consequences, or one who avoids a pain that produces no resultant pleasure? On the other hand, we denounce with righteous indignation and dislike men who are so beguiled and demoralized by the charms of pleasure of the moment, so blinded by desire, that they cannot foresee the pain and trouble that are bound to ensue.