Lista 6 i 7

Stwórz **uproszczoną** wersję gry w ruletkę. Menu zakładu:

Źródło: https://pl.wikipedia.org/wiki/Ruletka_(gra)

Zasady:
Użytkownik przy każdej turze wprowadza zakład w postaci – wersja uproszczona (format nazwa – wygrana):
- numer pojedynczy – 35:1
- trzy numery – 11:1
- kolumna – 2:1
- tuziny: 1 – 12, 13 – 24, 25 – 36 – 2:1
- parzyste, nieparzyste – 1:1
- numery 1 – 18 i 19 – 36 – 1:1
- czerwone, czarne – 1:1

Wybór ten ma być dokonywany przez kliknięcie/dotknięcie odpowiedniego elementu Menu.
Dodatkowo w każdej turze użytkownik wprowadza liczbę określającą wartość zakładu (załóżmy, że na początku użytkownik ma 100 kredytów).

Losujemy liczbę z przedziału od 0 do 36 i gdy wylosowana wartość zgadza się z dokonanym wyborem obliczamy jego wygraną dodając odpowiednią wartość wygranej do konta użytkownika. Gdy użytkownik przegrał, obstawiana kwota jest pobierana z jego konta.

Funkcjonalność aplikacji:

Lista 6:

1. Należy wyświetlić na stronie Menu zakładu w postaci przedstawionej na rysunku (wygląd koncepcyjny, można zrobić to lepiej).
2. Po kliknięciu odpowiedniej pozycji (zakładamy w uproszczonej wersji, że użytkownik może dokonać tyko pojedynczego zakładu) wybór użytkownika ma być pamiętany w programie (wyświetl odpowiednią informację o wyborze w konsoli)
3. Należy dostarczyć komponent wizualny, który pozwoli na wprowadzenie kwoty zakładu. Kwota zakładu ma być pamiętana w programie (wyświetl obstawioną kwotę w konsoli po jej wczytaniu)
4. Należy oprogramować logikę gry dla wyboru pojedynczej liczby (zakład 35:1) – funkcjonalność może ograniczyć się do wyświetlania komunikatu „Wygrana” lub „Przegrana” w zależności od wylosowanej liczby (nie trzeba na liście 6 oprogramować logiki związanej z bieżącym kontem użytkownika)
5. Dostarcz odpowiednie reguły CSS, które pozwolą w sposób odpowiedni prezentować wygląd graficzny na urządzeniu mobilnym (zastosuj odpowiednie @media w CSS)

Sposób oceniania:
1+2 DST
1+2+3 DST+
1+2+3+4 DB
1+2+3+4+5 BDB

Lista 7:

1. Oprogramowanie pełnej logiki związanej z zakładami: Wyświetlić wylosowaną wartość, Wyświetlić informację o obstawianej opcji przez użytkownika, wyświetlić informację o wygranej lub przegranej po każdej rundzie
2. Oprogramowanie logiki stanu konta użytkownika – przy wygranej odpowiednia liczba jednostek ma być doliczana do konta użytkownika, przy przegranej, obstawiona kwota ma być odejmowana od konta użytkownika. Przy zerowym lub ujemnym stanie konta gra ma się zakończyć z wyświetleniem informacji ile rund użytkownik był w stanie zaliczyć
3. Dodać animację, pokazującą kręcące się koło rulety. Ruleta ma się zatrzymać na polu, które zostało wylosowane w bieżącej rundzie (można przedstawić proces na zasadzie koła fortuny, gdzie wskaźnik określa wylosowaną wartość).

Sposób oceniania:
Pełna funkcjonalność listy 6 (może być bez punktu 5, jeżeli ktoś go nie zrealizował na liście 6) + 1 z listy 7 – DST
Poprzednie + 2 z listy 7 – DB
Poprzednie + 3 z listy 7 – BDB

Uwagi/Podpowiedzi:
- Zastosowanie w CSS elementu **`transform: rotate(…)`** pozwala na obrót elementu HTML (to nie jest jeszcze lista, na której powinno używać się elementu CANVAS z HTML5).

- Ze względu, na fakt że zadanie celowo jest nieco dłuższe (na 2 zajęcia), warto zastanowić się nad wykorzystaniem modułów przy tworzeniu kodu JS, bo będzie go sporo,
- Zadanie zostało celowo podzielone na 2 zajęcia – proszę nie przynosić całości na raz, bo projekt ma na celu wymusić kontynuowanie zagadnienia po pewnym odstępie czasowym (niekiedy użycie pewnych mechanizmów, dobrego sposobu nazewnictwa zmiennych, dobrych praktyk programowania widać, gdy do kodu trzeba siadać drugi raz). Treści obu list są prezentowane od razu tylko po to, żeby była widoczna logika całości.
- Ze względu na dużą liczbę elementów, które może kliknąć użytkownik warto zastanowić się nad delegowaniem obsługi zdarzeń do przodka elementów klikalnych (można wykorzystać w znacznikach przycisków atrybuty poprzedzone prefixem data-, do przechowywania w DOM informacji na temat danych bieżącego elementu),
- Warto zastanowić się nad generowaniem własnych zdarzeń przy obsłudze logiki gry (np. przy wygranej jeden typ zdarzenia, przy przegranej inny typ zdarzenia),
- Warto sprawdzić, czy zdarzenia przypisane dla wersji desktop będą odpowiednio działały na urządzeniu mobilnym, dotykowym (nie dla wszystkich obiektów DOM wszystkie zdarzenia na różnych urządzeniach działają tak samo),
- Starać się stworzyć interfejs użytkownika, który będzie jak najbardziej estetyczny i intuicyjny (elementy mają być czytelne, odpowiednio duży tekst – szczególnie na urządzeniu mobilnym elementy klikalne powinny być duże, dobrać dobry kontrast, elementy wybrane/aktywne powinny być w jakiś sposób wyróżnione – np. dokonany wybór zakładu – dostarczony obrazek menu jest tylko wyglądem koncepcyjnym).
- Stworzyć program maksymalnie zabezpieczony przed błędami użytkownika (np. walidacja pola zakładu).
- Żeby poćwiczyć zagadnienia związane z HTML i CSS koło rulety warto wykonać samemu (posługując się tylko elementami HTML i CSS), ale jeżeli komuś będzie brakowało czasu, może również użyć gotowej grafiki (nie zostanie obniżona ocena z tego powodu).
- Jako inspiracja dla procesu kręcenia ruletą podaję link:
https://codepen.io/pder/pen/WYrRQm
(oczywiście aplikacja na zajęcia nie musi wyglądać tak dobrze, ważne żeby każdy sam wykonał kod na tyle na ile umie).