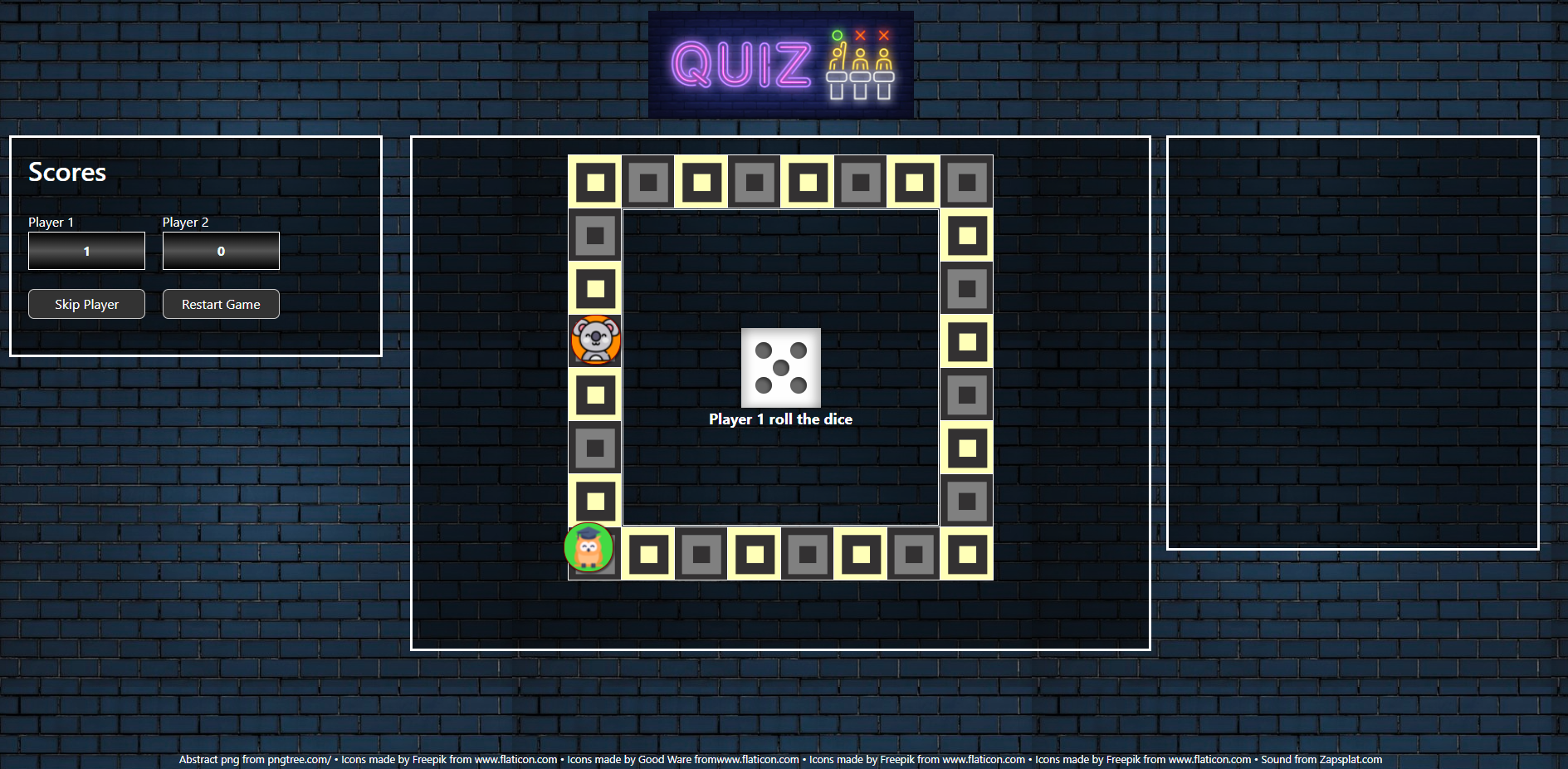
# Παιχνίδι τύπου Trivial pursuit – Τεχνική Αναφορά

Το συγκεκριμένο παιχνίδι εκπαιδευτικού σκοπού είναι εμπνευσμένο από το κλασσικό παιχνίδι Trivial pursuit, και αποτελεί μια προσπάθεια εξέλιξής του με προσθήκη διάφορων οπτικοακουστικών πολυμέσων και επιλογής βαθμού δυσκολίας. Αναπτύχθηκε με χρήση HTML5, JavaScript και CSS.



## Διάρθρωση αρχείων

Η οργάνωση των αρχείων έχει ως εξής:

* Φάκελος **audio**: περιέχονται τα αρχεία ήχου που έχουν συνδεθεί με την κίνηση του κάθε avatar, το αρχείο ήχου που έχει συνδεθεί με την κίνηση του ζαριού, καθώς και το αρχείο ήχου που έχει συνδεθεί με την νίκη και τη λήξη του παιχνιδιού
* Φάκελος **img**: περιέχονται όλα τα πολυμέσα εικόνας που χρησιμοποιήθηκαν (avatars, board, εικόνα background, εικόνα logo)
* Φάκελος **css**: περιέχονται αρχεία στυλ CSS του παιχνιδιού
* Φάκελος **js**: περιέχονται αρχεία JavaScript για την “online” λειτουργία του παιχνιδιού
* Αρχείο **app.js**: το πιο σημαντικό αρχείο JavaScript της εφαρμογής στο οποίο έχουν υλοποιηθεί οι βασικές λειτουργικότητες
* Αρχείο **index.html**: βασικό αρχείο HTML5 της εφαρμογής για την συνολική εμφάνιση του παιχνιδιού, στο οποίο έχει δοθεί ιδιαίτερη προσοχή στην αισθητική καθώς θεωρείται βασική προϋπόθεση επιτυχίας των παιχνιδιών. Επίσης έχουν προστεθεί ως αναφορές, οι πηγές πολυμέσων που χρησιμοποιήθηκαν στο footer.
* Αρχείο **style.css**: περιέχει το custom style

## Υλοποίηση λειτουργικότητας ερωτήσεων

Για την υλοποίηση του τμήματος ερωταπαντήσεων του παιχνιδιού, συνδέθηκε σχετικό API. Xρησιμοποιήθηκε επίσης η JQuery (έκδοση 2.1.4) για να βοηθήσει στη ροή του παιχνιδιού. Για τον λόγο αυτό, δημιουργήθηκε και το container ερωτήσεων στο index.html.

Η εναλλαγή ερωτήσεων έχει συνδεθεί με την κίνηση του εκάστοτε avatar στο board.

Κώδικας και μεταβλητές προστέθηκαν στο app.js για να επιτρέψουν επιλογή διαφορετικών επίπεδων δυσκολίας με τις μεταβλητές difficultyLevel (easy, medium και hard) και scoreJump (1, 2, ή 3 βάσει του difficultyLevel). Προστέθηκε ο σχετικός κώδικας στο ερώτημα API για να υποδείξει το επίπεδο δυσκολίας.

Για να διασφαλιστεί ότι το παιχνίδι δεν θα επαναλαμβάνει τις ίδιες ερωτήσεις με την ίδια σειρά κάθε φορά που κάποιος θα παίζει (μη γραμμικότητα), ορίστηκε η ανάκτηση ερωτήσεων απο το API να επιτρέπει τη λήψη 50 ερωτήσεων ανά παρτίδα και η προβολή μίας με τυχαίο τρόπο.

Τέλος, αναπτύχθηκε κώδικας για τη λήψη επιβεβαίωσης σε περίπτωση που οι παίκτες επιλέξουν να συνεχίσουν να παίζουν σε νέα παρτίδα μέσω του κουμπιού επανεκκίνησης "Restart Game", ο οποίος επανεκκινεί τη σελίδα και τα variable states.

## Ροή του παιχνιδιού

Τα δύο κύρια αρχεία που ρυθμίζουν τη ροή παιχνιδιού είναι τα **app.js** και **index.html.** Πιο συγκεκριμένα:

* Η κλάση initial-body περιέχει ουσιαστικά όλη τη σελίδα. Το παιχνίδι ξεκινάει με το τμήμα της αρχικής επιλογής αριθμού παικτών, ονόματος παίκτη και avatar, επίπεδο δυσκολίας και το κουμπί Play. Αφού ο χρήστης κλικ στο κουμπί με τo tag “play-clicked”, η σελίδα αλλάζει το “start-game” tag. Αυτό το στοιχείο βρίσκεται στο app.js
* Στο start game βρίσκονται το board, o πίνακας βαθμολογίας, το ζάρι, τα πιόνια/avatars, το και το container ερωτήσεων. Τα βασικά στοιχεία που περιέχονται είναι τα παρακάτω:

- Το Div με το id “board” περιέχει το table με το ταμπλώ που κινούνται οι παίκτες. Τα Td με το id του ταμπλό είναι για τις κινήσεις των player-cards (που ως οντότητες περιέχουν το όνομα, το score και το επιλεγμένο avatar).

- To Div με το score-board που περιέχει τα αποτελέσματα βάσει των σωστών απαντήσεων των παικτών. Συγκεκριμένα τη Βαθμολογία, και τα info-header και info-body. Στην εφαρμογή js, η συνάρτηση score(boolean) ελέγχει τη βαθμολογία του currentPlayer.

- Το Dice περιέχει το animation των tags του ζαριού.

- Τα Pieces είναι οι κάρτες των παικτών. Με την εκάστοτε κίνηση κάποιου piece, ενεργοποιείται η getQuestion() στο app.js και οι ετικέτες του κοντέινερ ερωτήσεων αλλάζουν.

- Το container ερωτήσεων περιέχει, συνδέεται και αποφασίζει την εναλλαγή παικτών και του σκορ. Το κοντέινερ ερωτήσεων λαμβάνει μια ερώτηση από τo <https://opentdb.com/api.php?amount=1&type=multiple> και το api στέλνει ένα response όπως π.χ. το παρακάτω:

{

"response\_code": 0,

"results": [

{

"category": "Entertainment: Video Games",

"type": "multiple",

"difficulty": "medium",

"question": "From where does the Nintendo video game character Mario derive his name?",

"correct\_answer": "The landlord of the American headquarters",

"incorrect\_answers": [

"Shigeru Miyamoto&#039;s father in law",

"Satoru Iwata&#039;s plumber",

"The date (Mar 10) that Donkey Kong was released."

]

}

]

}

Με το στοιχείο results[0], προετοιμάζονται 4 ετικέτες κουμπιών. 3 από αυτές είναι οι λάθος απαντήσεις. Οι ετικέτες correct\_answers ορίζουν τα κουμπιά σε σχέση με το αν είναι οι σωστές ή οι λάθος. Στη συνέχεια, στον πίνακα, ανακατεύονται έτσι ώστε η σωστή απάντηση να είναι σε διαφορετική θέση κάθε φορά.

Όταν πατηθεί η σωστή απάντηση, καλείται η μέθοδος correctClick(buttonNumber):

- Αλλάζει το χρώμα του container της απάντησης σε πράσινο

- Το σκορ του τρέχοντος παίκτη αυξάνεται και προχωράει το state σε “Roll the dice”

Όταν επιλεχθεί λάθος απάντηση, καλείται η μέθοδος wrongClick(buttonNumber, correctButtonNumber)

- Αλλάζει το χρώμα του κουμπιού της απάντησης σε κόκκινο και με πράσινο φαίνεται η σωστή απάντηση

- To state αλλάζει σε “next player”.

Και οι δύο αυτές συναρτήσεις (correctClick() και wrongClick() ) χωρίστηκαν σε δύο τμήματα (π.χ. correctClick1() και correctClick2() ) ουτωσώστε να μεσολαβήσει ένα setTimeout που δίνει λίγο χρόνο για να φανεί ο εκάστοτε χρωματισμός απάντησης (πράσινο/κόκκινο) και έπειτα να καλέσει το δεύτερο τμήμα της κάθε συνάρτησης για να προχωρήσει το state.

Έχει προστεθεί επίσης κώδικας ώστε να διασφαλίζεται ότι το παιχνίδι και τα states αρχικοποιούνται πλήρως κατα την έναρξη του παιχνιδιού.Προστέθηκε και κουμπί “Skip Player” που ενεργοποιεί την συνάρτηση skipPlayer() και πηγαίνει την σειρά στον επόμενο, όταν κάποιος παίκτης φέρει ζαριά που τον αναγκάζει να μετακινηθεί σε θέση δεσμευμένη απο κάποιον συμπαίκτη.Επίσης, θέσαμε τα default ονόματα παικτών ως «Player 1», «Player 2» κλπ όταν δεν εισάγεται κάποιο όνομα. Τέλοις, η συνάρτηση doWin(playerName) χειρίζεται την εμφάνιση του banner νίκης και τον ήχο.

**Αναφορές:**

Χρησιμοποιήθηκε κώδικας απο την εξής πηγή για την μετακίνηση drag-and-drop των εικονιδίων - avatars: <https://github.com/bug7a/iconselect.js>