



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΕΦΑΡΜΟΣΜΕΝΩΝ ΜΑΘΗΜΑΤΙΚΩΝ & ΦΥΣΙΚΩΝ ΕΠΙΣΤΗΜΩΝ

**Το πανηγυρι των σωματιδίων:
Μια εκπαιδευτική διαδραστική εφαρμογή για την εξοικείωση με τα
στοιχειώδη σωματίδια και τις αλληλεπιδράσεις μεταξύ τους**

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ
του
ΛΑΛΟΥ ΘΩΜΑ

Επιβλέπων Καθηγητής: Τράκας Νικόλαος
(Καθηγητής Ε.Μ.Π)

Αθήνα, Σεπτέμβριος 2021

ΛΑΛΟΣ ΘΩΜΑΣ

Διπλωματούχος Σχολής Εφαρμοσμένων Μαθηματικών & Φυσικών Επιστημών Ε.Μ.Π

Copyrights © Λάλος 2021 – All Rights Reserved

Περίληψη

Από τα αρχαία χρόνια η ανθρωπότητα προσπαθούσε να βρει απαντήσεις στο ερώτημα από τι δομείται η ύλη. Από την αρχική απόπειρα του Λέυκιππου και του μαθητή του Δημόκριτου σχετικά με το άτομο μέχρι τα τεράστια πειράματα στοιχειωδών σωματιδίων τα οποία πραγματοποιούνται στο CERN κι άλλα ερευνητικά κέντρα ανά τον κόσμο έχουν περάσει χιλιάδες χρόνια ερευνών, πειραμάτων και μελετών. Η σωματιδιακή φυσική, όμως, πέρα από τις απαντήσεις σε βασικά ερωτήματα που απασχολούν την ανθρωπότητα αιώνες έχει δώσει λύσεις και σε ζητήματα της καθημερινότητας μέσω της ανάπτυξης τεχνολογιών ικανών να ικανοποιήσουν ανθρώπινες ανάγκες. Αυτή η κατάσταση δημιουργεί την ανάγκη για κατανόηση των βασικών στοιχείων της δομής της ύλης από την κοινωνία και κυρίως μέσα από την εκπαιδευτική διαδικασία με μεθόδους που θα δώσουν την ευκαιρία να γνωρίσουν οι μαθητές τον κόσμο της σωματιδιακής φυσικής. Το “πανηγύρι των σωματιδίων” εξυπηρετεί αυτή ακριβώς την ανάγκη, ως μία διαδραστική εφαρμογή στην οποία οι μαθητές/τριες θα έχουν την ευκαιρία μέσω μιας απλής διαδικασίας εφαρμογής της θεωρίας των στοιχειωδών σωματιδίων και των αλληλεπιδράσεών τους να δημιουργήσουν μία οπτικοποίηση της γνώσης τους χωρίς να είναι ανάγκη να εισέλθουν σε έναν κόσμο δύσκολων εξισώσεων και δυσνόητων εννοιών.

Περιεχόμενα

Γνωριμία με τα στοιχειώδη σωματίδια

1.Εισαγωγή κι επεξήγηση της εφαρμογής.....	1
2.Παρουσίαση βασικής θεωρίας	1
3.Οδηγίες χρήσης κι εισαγωγή στον κώδικα που χρησιμοποιήθηκε.....	4
4. Κώδικας εφαρμογής.....	4

Αλληλεπιδράσεις Στοιχειωδών Σωματιδίων

1. Εισαγωγή κι επεξήγηση της εφαρμογής.....	12
2. Παρουσίαση βασικής θεωρίας.....	13
3. Οδηγίες χρήσης κι εισαγωγή στον κώδικα που χρησιμοποιήθηκε.....	14
4. Κώδικας Εφαρμογής.....	14

ΓΝΩΡΙΜΙΑ ΜΕ ΤΑ ΣΤΟΙΧΕΙΩΔΗ ΣΩΜΑΤΙΔΙΑ

ΕΙΣΑΓΩΓΗ

Το πανηγυρι των σωματιδίων είναι μία διαδραστική εφαρμογή με σκοπό την εξοικείωση των μαθητών με τα στοιχειώδη σωματίδια και τις αλληλεπιδράσεις που συναντώνται μεταξύ τους. Περιλαμβάνει δύο εφαρμογές, η πρώτη σχετίζεται με την δομή των στοιχειωδών σωματιδίων και την σύνθεση μέσω των quark που τα απαρτίζουν και η δεύτερη σχετικά με τις αλληλεπιδράσεις όπως αυτές απεικονίζονται μέσω των διαγραμμάτων Feynman.

Για εισαγωγή σχετικά με τα στοιχειώδη σωματίδια μπορούμε να παραθέσουμε τα κείμενα που έχουν συμπεριληφθεί και στην ίδια την εφαρμογή:

Πίνακας Στοιχειωδων Σωματιδιων

Το άτομο

Η ατομική δομή της ύλης, ως φιλοσοφική ιδέα τουλάχιστον, γνωρίζουμε ότι ξεκίνησε από τον Λέυκιππο και τον Δημόκριτο (4ος αιώνας π.Χ). Πέρασαν πολλοί αιώνες, ώσπου τον 18ο αιώνα ο Λαβουαζιέ (Antoine Laurent Lavoiser), μαζί με μία ολόκληρη σειρά άλλων επιστημόνων, επιβεβαίωσαν με πειράματά τους την ορθότητα αυτής της άποψης.

Πυρήνας και Ηλεκτρόνια

Στις αρχές του 20ου αιώνα, ο Ράδερφορντ (Ernest Rutherford) κατάφερε να δείξει ότι το άτομο έχει ένα πολυσυμπαγή, όσο και σχετικά πολύ μικρό πυρήνα, ενώ ο υπόλοιπος χώρος του ατόμου φαίνεται κενός. Το βασικό πείραμα που πραγματοποίησε ήταν ο βομβαρδισμός ενός λεπτού φύλλου χρυσού με σωματίδια α (που σήμερα γνωρίζουμε ότι είναι πυρήνες του στοιχείου Ήλιο, He) πολύ μεγάλης ταχύτητας. Ενώ λοιπόν τα πιο πολλά από αυτά τα σωματίδια διαπερνούσαν το φύλλο με πολύ μικρές αποκλισεις, μερικά “γύριζαν” προς τα πίσω.

Όπως έλεγε και ο ίδιος ο Ράδερφορντ, “...ήταν το ίδιο απίστευτο σαν να είχες πυροβολήσει ένα τσιγαρόχαρτο με ένα βλήμα 1 ιντσών και το βλήμα να γύριζε κατευθείαν πίσω και να σε χτυπούσε”. Ο πυρήνας του ατόμου είναι θετικά φορτισμένος ενώ, κατά τον Ράδερφορντ, ένα πλήθος (αρνητικά φορτισμένων) ηλεκτρονίων γέμιζε τον κενό χώρο.

Έπρεπε να έρθει ο Μπορ (Niels Bohr) για να βάλει τα ηλεκτρόνια στην “θέση” τους, δηλαδή να περιστρέφονται σε κυκλικές ή ελλειπτικές τροχιές γύρω από τον πυρήνα (το ηλεκτρόνιο είχε ήδη ανακαλυφθεί από τον Τόμσον (Joseph John Thomson) λίγα χρόνια πριν).

Λίγα χρόνια μετά, η κβαντική θεωρία βρίσκεται ήδη σε εξέλιξη, η διττή φύση του φωτός (κυματική και σωματιδιακή) θεωρείται δεδομένη, και ο Ντε Μπρόγι, (Louis De Broglie) εισάγει την ιδέα ότι και η ύλη μπορεί να επιδειξει κυματική συμπεριφορά, υπόθεση που επιβαιώνεται και πειραματικά. Έτσι, από το 192 περίπου και μετά, αλλάζει εντελώς ο τρόπος που “βλέπουμε” τα στοιχειώδη σωματίδια, που δεν μπορούν να περιγραφούν πια ως υλικά σημεία με συγκεκριμένη θέση στο χώρο, αλλά ως κυματοσυναρτήσεις με πιθανότητα θέσης σε δεδομένη χρονική στιγμή.

Πρωτόνια και Νετρόνια

Νέα πειράματα, τη 2η και 3η δεκαετία του αιώνα μας, έδωσαν την νέα πληροφορία, ότι ο πυρήνας του ατόμου σχηματίζεται από δύο ειδών σωματίδια: το πρωτόνιο, με θετικό ηλεκτρικό φορτίο (ίσο κατ'απόλυτη τιμή με αυτό του ηλεκτρονίου) και το νετρόνιο, που έχει σχεδόν την ίδια μάζα με το πρωτόνιο αλλά είναι ηλεκτρικά ουδέτερο.

ΟΙ ΓΕΝΙΕΣ ΤΩΝ ΣΩΜΑΤΙΔΙΩΝ

Η 1η ΓΕΝΙΑ

Με αυτά τα στοιχεία τώρα μπορούμε να περιγράψουμε όλη τη φύση, εμάς τους ίδιους, το σύμπαν ολόκληρο. Πράγματι, η σημερινή επιστήμη της δομής της ύλης, η φυσική των στοιχειώνδων σωματιδίων, χρειάζεται μόνο 4 βασικούς λίθους για να χτίσει τον σημερινό κόσμο. Το ανώ και κάτω κούαρκ, το ηλεκτρόνιο κι επιπλέον ένα πολύ παράξενο σωματίδιο, το νετρίνο. Αυτό το τελευταίο είχε προβλεφθεί από τον Παουλί(Wolfgang Pauli) από το 1930, αλλά μόνο το 1956 αποδείχτηκε πειραματικά η ύπαρξη του.

Είναι ένα ουδέτερο ηλεκτρικά σωματίδιο, με εκπληκτικά μικρή μάζα (στην πραγματικότητα η μάζα του είναι μικρότερη από το σφάλγμα που κάνουν οι πειραματικές συσκευές και θεωρητικά το αντιμετωπίζουμε σαν να μην έχει μάζα, κάτι τελείως αποδεκτό από την θεωρία της σχετικότητας). Το νετρίνο παρουσιάζεται σαν προϊόν διάσπασης του νετρονίου. Το ηλεκτρόνιο και το νετρίνο ονομάζονται λεπτόνια(από τη λέξη "λεπτός"). Στην πρώτη γραμμή του παρακάτω πίνακα, φαίνονται ακριβώς οι τέσσερις αυτοί δομικοί λίθοι της φύσης με τα αντίστοιχα ηλεκτρικά φορτία τους.

Η 2η και η 3η Γενιά

Παρ'όλο που τα τέσσερα αυτά σωματίδια σχηματίζουν την ύλη όλου του σημερινού σύμπαντος, στα πειράματα που γίνονται (και, όπως πιστεύουμε, επίσης κατά τα πρώτα δευτερόλεπτα της δημιουργίας του σύμπαντος), εμφανίζονται και άλλα σωματίδια που έχουν όμως εκπληκτικές ομοιότητες με τα τέσσερα που προαναφέραμε και που θα ονομάζουμε 1η γενιά. Πράγματι λοιπόν, εμφανίζονται άλλες 2 γενιές σωματιδίων, η καθεμιά με 2 λεπτόνια και 2 κουάρκ. Στην 2η γενιά έχουμε το σωματίδιο μ (μυόνιο) και το δικό του νετρίνο (διαφορετικό από αυτό της 1ης γενιάς), ως λεπτόνια, καθώς και το γοητευτικό (charm) και το παράξενο (strange) κουάρκ με φορτία ακριβώς αντίστοιχα με τα σωματίδια της 1ης γενιάς (βλέπε τον παραπάνω πίνακα). Οι μάζες των σωματιδίων της 2ης γενιάς είναι μεγαλύτερες από τις αντίστοιχες μάζες της 1ης. Στην 3η γενιά, εμφανίζονται το σωματίδιο τ (ταυ) και το δικό του νετρίνο ως λεπτόνια, και το υψηλό (top) και το χαμηλό (bottom) κουάρκ (κατ'άλλους κορυφή και πυθμένας). Και πάλι, η 3η γενιά εμφανίζεται με ακόμα μεγαλύτερη μάζα. Είναι γεγονός ότι η ύπαρξη των 3 γενεών, με πανομοιότυπα χαρακτηριστικά, αποτελεί ένα από τα αναπάντητα ακόμα ερωτήματα της Φυσικής των Στοιχειωδών Σωματιδίων. Πάντως, τα πειράματα συνηγορούν στην ύπαρξη τριών κι όχι περισσότερων γενεών.

Κουάρκ

Στην δεκαετία του εξήντα και του '70 έγινε κατανοητά, αρχικά θεωρητικά και κατόπιν επιβεβαιώθηκε και πειραματικά, ότι τα πρωτόνια και τα νετρόνια δεν είναι στοιχειώδη αλλά έχουν εσωτερική δομή: αποτελούνται δηλαδή από μικρότερα σωματίδια που ονομάστηκαν κουάρκ (quark). Τα κουάρκ παρουσιάζονται σε δύο είδη: το άνω (up) κουάρκ και το κάτω (down) κουάρκ. Τα κουάρκ είναι ηλεκτρικά φορτισμένα: το άνω κουάρκ έχει φορτίο $2/3$, ενώ το κάτω κουάρκ έχει φορτίο $-1/3$. Τα σύνθετα σωματίδια που αποτελούνται από κουάρκ, όπως το πρωτόνιο και το νετρόνιο, ονομάζονται αδρόνια. Το πρωτόνιο αποτελείται από 2 άνω κουάρκ και 1 κάτω κουάρκ ενώ το νετρόνιο από 2 κάτω κουάρκ κι 1 άνω κουάρκ.

... και τα αντισωματίδια

Επομένως, θα μπορούσαμε να πούμε ότι συνολικά 12 σωματίδια αποτελούν τους βασικούς δομικούς λίθους τους σύμπαντος, από την γέννησή του μέχρι σήμερα; Όχι ακριβώς! Η σύγχρονη θεωρία που περιγράφει τον μικρόκοσμο, η κβαντομηχανική, προβλέπει την ύπαρξη του αντισωματιδίου: σε κάθε σωματίδιο αντιστοιχεί άλλο ένα, με ίδια μάζα και αντίθετο ηλεκτρικό φορτίο. Έτσι, ο συνολικός αριθμός των σωματιδίων διπλασιάζεται. Η ύπαρξη των αντισωματιδίων προβλέφθηκε θεωρητικά από τον Άγγλο θεωρητικό Φυσικό Ντιράκ (Dirac) την δεκαετία του '30. Η ανακάλυψη του αντιηλεκτρονίου (ποζιτρόνιο) έδωσε τον πρώτο θρίαμβο της θεωρίας του Ντιράκ. Σήμερα, η ύπαρξη αντισωματιδίου για κάθε σωματίδιο έχει αποδειχθεί πειραματικά και αποτελεί έναν από τους ακρογωνιαίους λίθους της θεωρίας των Στοιχειωδών Σωματιδίων. Παρακάτω βλέπετε έναν πίνακα με τα στοιχειώδη σωματίδια.

Από τα κουάρκ στα ανδρόνια

Τα ανδρόνια είναι σύνθετα σωματίδια τα οποία αποτελούνται από κουάρκ. Η θεωρία που διέπει τις αλληλεπιδράσεις των κουάρκ (κβαντική χρωμοδυναμική) προβλέπει ποιοι είναι οι επιτρεπτοί συνδυασμοί των κουάρκ από τα οποία αποτελούνται τα ανδρόνια. Τα βαρυόνια (πρωτόνιο, νετρόνιο κ.λ.π) σχηματίζονται από συνδυασμό 3 κουάρκ, ενώ τα μεσόνια (καόνιο, σωματίδιο π κλπ) σχηματίζονται από τρία αντικουάρκ. Περισσότερα για τα ανδρόνια (βαρυόνια και μεσόνια) και για τις διαφορές τους με τα λεπτόνια θα πούμε αργότερα, στην μελέτη των Αλληλεπιδράσεων.

Στοιχειώδη Σωματίδια

Η πρώτη διαδραστική εφαρμογή έχει στόχο να εξοικειωθεί ο μαθητής με την δομή των στοιχειωδών σωματιδίων. Συγκεκριμένα αφού έχει μελετήσει βασικά στοιχεία της θεωρίας στοιχειωδών σωματιδίων του δίνεται μία λίστα με όλα τα quarks και από αυτά διαλέγει τους κατάλληλους συνδυασμούς οι οποίοι οδηγούν στην σύσταση ενός σωματιδίου. Όταν ο μαθητής δεν επιλέξει τον σωστό συνδυασμό τότε βλέπει στην οθόνη του το μήνυμα “δυστυχώς η συσχέτιση των σωματιδίων δεν είναι σωστή, παίξε σωστά πατώντας επαναφορά”.

Το συγκεκριμένο παιχνίδι δεν έχει κάποιο χρονικό όριο ή κάποιο όριο βημάτων με αποτέλεσμα ο μαθητής να έχει τον χρόνο και την ευκαιρία να καταπιαστεί ουσιαστικά και να κατανοήσει την θεωρία σε μία μορφή πρακτικής άσκησης.

Για την πραγματοποίηση του κώδικα έγινε χρήση του Hook “useState” της React. Για αρχή ορίζουμε τους σωστούς συνδυασμούς ως τα σωστά states από τις επιλογές του χρήστη και ορίζουμε την ταυτότητα του σωματιδίου που προκύπτει από τον κάθε συνδυασμό.

```
import React, { useState } from 'react';
import ParticleButton from './elemParticleGameButton'
import Particle from './elemParticle'
```

```
const correctParticleCombinations = {
  // FOR_TWO_PARTICLES: [
  //   ["u", "u_"],
  //   ["d", "d_"],
  //   ["s", "s_"],
  //   ["u", "d_"],
  //   ["u", "s_"],
  //   ["d", "u_"],
  //   ["d", "s_"],
  //   ["s", "u_"],
  //   ["s", "d_"],
  // ],
```

FOR_TWO_PARTICLES: [

[["u", "u_", <text>η ή π⁰ μεσόνοιο</text>],
[["d", "d_", <text>η ή π⁰ μεσόνοιο</text>],
[["s", "s_", <text>η μεσόνοιο</text>],
[["u", "d_", <text>π⁺ μεσόνοιο</text>],
[["u", "s_", <text>Κ⁺ μεσόνοιο</text>],
[["d", "u_", <text>π⁻ μεσόνοιο</text>],
[["d", "s_", <text>Κ⁰ μεσόνοιο</text>],
[["s", "u_", <text>Κ⁻ μεσόνοιο</text>],
[["s", "d_", <text>Κ⁰

μεσόνοιο</text>],

],

// FOR_THREE_PARTICLES: [

// ["u", "u", "u"],
// ["u", "u", "d"],
// ["u", "u", "s"],
// ["u", "d", "d"],
// ["u", "s", "s"],
// ["d", "d", "d"],
// ["d", "d", "s"],
// ["s", "s", "s"],
// ["s", "s", "d"]

//]

FOR_THREE_PARTICLES: [

[["u", "u", "u"], <text>Δ⁺</text>],
[["u", "u", "d"], <text>ρ πρωτόνοιο</text>],
[["u", "u", "s"], <text>Σ⁺</text>],
[["u", "d", "d"], <text>η νετρόνοιο</text>],
[["u", "s", "s"], <text>Ξ⁰</text>],
[["d", "d", "d"], <text>Δ⁻</text>],
[["d", "d", "s"], <text>Σ⁻</text>],
[["s", "s", "s"], <text>Ω⁻</text>],
[["d", "s", "s"], <text>Ξ⁻</text>],
[["u", "d", "s"], <text>Λ ή Σ⁰</text>],
[["u_", "u_", "u_"], <text>Δ⁻⁻</text>],
[["u_", "u_", "d_"], <text>ρ

αντιπρωτόνοιο</text>],

[["u_", "u_", "s_"], <text>Σ⁻</text>],

[["d_", "d_", "d_"], <text>Δ⁺</text>],

[["u_", "d_", "d_"], <text>η

αντινετρόνοιο</text>],

[["d_", "d_", "s_"], <text>Σ⁺</text>],

[["s_", "s_", "s_"], <text>Ω⁺</text>],

[["u_", "s_", "s_"], <text><span style={{ 'textDecoration':

```

'overline' }}>Ξ</span><sup>0</sup></text>],
  [{"d_", "s_", "s_"}, <text>Ξ<sup>+</sup></text>],
  [{"d_", "s_", "s_"}, <text>Ξ<sup>+</sup></text>],
  [{"u_", "d_", "s_"}, <text><span style={{ 'textDecoration': 'overline' }}>Λ</span> ή <span
style={{ 'textDecoration': 'overline' }}>Σ</span><sup>0</sup></text>]
  ]
};

```

Μέσω της `useState` ορίζουμε τις καταστάσεις στις οποίες ο χρήστης έχει επιλέξει σωματίδια, έχει επιλέξει παραπάνω σωματίδια απ'οτι ορίζεται από τους κανόνες, το τέλος του παιχνιδιού και τον σωστό συνδυασμό και στην συνέχεια μέσω μιας `if function` ορίζουμε κάθε πιθανή κίνηση στην οποία θα προβεί ο χρήστης.

Ορίζουμε την `function Particle Button` έτσι ώστε κάθε φορά ο χρήστης να πατάει το “παίξε” και να ξεκινάει η διαδικασία του παιχνιδιού και την `ResetGame` για κάθε φορά που ο χρήστης θα ξεκινάει το παιχνίδι από την αρχή.

```

const ElemParticlesGame = () => {

  const [selectedParticles, setSelectedParticles] = useState([]); //
  const [warningTooMany, setWarningTooMany] = useState(false);
  const [endGame, setEndGame] = useState(false);
  const [winner, setWinner] = useState(false);
  const [correctParticleCombinationFound, setCorrectParticleCombinationFound] =
  useState("");

  const addNewParticle = (particleName) => {
    const newSelectedParticles = selectedParticles.slice();

    if (selectedParticles.length <= 2) {
      newSelectedParticles.push(particleName);
      setSelectedParticles(newSelectedParticles);
    } else {
      if (!endGame) {
        setWarningTooMany(true);
      }
    }
  };

  const checkDoubleParticleCombination = () => {
    const selectedParticlesJoined = selectedParticles.slice().sort().join("");
    let selectedMatchIsCorrect = -1;
    let particlesMatched = false;

```

```

// object που ορίζει τους σωστούς συνδυασμούς για μετεπειτα συγκριση
correctParticleCombinations.FOR_TWO_PARTICLES.forEach(correctParticleCombination =>
{
    if (!particlesMatched) {
        const correctParticleCombinationJoined =
correctParticleCombination[0].slice().sort().join("");
        selectedMatchIsCorrect =
correctParticleCombinationJoined.localeCompare(selectedParticlesJoined);

        if (selectedMatchIsCorrect === 0) {
            setWinner(true);
            setEndGame(true);
            particlesMatched = true;
            setCorrectParticleCombinationFound(correctParticleCombination[1]);
        }
    }
});

if (selectedMatchIsCorrect !== 0) {
    setWinner(false);
    setEndGame(true);
    setCorrectParticleCombinationFound("");
}
};

const checkTripleParticleCombinations = () => {
    const selectedParticlesJoined = selectedParticles.slice().sort().join("");
    let selectedMatchIsCorrect = -1;
    let particlesMatched = false;

correctParticleCombinations.FOR_THREE_PARTICLES.forEach(correctParticleCombination
=> {
    if (!particlesMatched) {
        const correctParticleCombinationJoined =
correctParticleCombination[0].slice().sort().join("");
        selectedMatchIsCorrect =
correctParticleCombinationJoined.localeCompare(selectedParticlesJoined);

        if (selectedMatchIsCorrect === 0) {
            setWinner(true);
            setEndGame(true);
            particlesMatched = true;
            setCorrectParticleCombinationFound(correctParticleCombination[1]);
        }
    }
}
}

```

```

});

    if (selectedMatchIsCorrect !== 0) {
        setWinner(false);
        setEndGame(true);
        setCorrectParticleCombinationFound("");
    }
    setWarningTooMany(false);
};

const checkParticleCombination = () => {
    if (selectedParticles.length === 2) {
        checkDoubleParticleCombination();
    } else if (selectedParticles.length === 3) {
        checkTripleParticleCombinations();
    }
};

```

Μέσω της function ResetGame δίνουμε την δυνατότητα στον χρήστη να ξαναπαίξει από την αρχή.

```

const resetGame = () => {
    setSelectedParticles([]);
    setWarningTooMany(false);
    setEndGame(false);
    setWinner(false);
    setCorrectParticleCombinationFound("")
};

```

Παρακάτω σχεδιάζουμε μέσω κώδικα την τελική διάδραση που θα έχει ο χρήστης με την εφαρμογή, τις κινήσεις των σωματιδίων προς το bowl και τα μυνήματα που θα του εμφανίζονται στην αρχική οθόνη.

//JSX συνδυασμος HTML με Javascript

```

return (
    <div className="homepage-container">
        <div className="particle-display">
            <div className="bowl"></div>
            {selectedParticles.map((particleName, index) => {
                return (
                    <Particle
                        name={particleName}
                        index={index}
                        key={index}
                    />
                )
            })}
        </div>
    </div>
);

```

```

{warningTooMany ? <div className="particles-error">Μπορείς να επιλέξεις μέχρι τρία
σωματίδια. Δοκίμασε ξανά πατώντας επαναφορά.</div> : null}
  {endGame ? winner ? <div className="particles-success"> Μπράβο, βρήκες το
{correctParticleCombinationFound}.<br />Παίξε ξανά πατώντας επαναφορά.</div> : <div
className="particles-error">Δυστυχώς η συσχέτιση των σωματιδίων δεν είναι σωστή, παίξε
ξανά πατώντας επαναφορά.</div> : null}
    <div className="particle-header">
      Επίλεξε τα σωματίδια και προσπάθησε να σχηματίσεις την σωστή σχέση,
επαληθεύοντας όσα έμαθες στην θεωρία.
    </div>
  </div>

```

```

<div className="bariums-mesons">
  <ParticleButton
    id="u"
    name="u"
    load="2/3"
    index="1"
    displayName={"u"}
    addnewparticle={addNewParticle}
  />
  <ParticleButton
    id="d"
    name="d"
    load="-1/3"
    index="2"
    displayName={"d"}
    addnewparticle={addNewParticle}
  />
  <ParticleButton
    id="s"
    name="s"
    load="-1/3"
    index="3"
    displayName={"s"}
    addnewparticle={addNewParticle} // arrowfuction
  />
  <ParticleButton
    id="u_"
    name="u_"
    load="-2/3"
    index="4"
    displayName={{"<span style={{ 'textDecoration': 'underline' }}>u</span>}}
    addnewparticle={addNewParticle}
  />

```

```

<ParticleButton
  id="d_"
  name="d_"
  load="1/3"
  index="5"
  displayName={<span style={{ 'textDecoration': 'underline' }}>d</span>}}
  addnewparticle={addNewParticle}
/>
<ParticleButton
  id="s_"
  name="s_"
  load="1/3"
  index="6"
  displayName={<span style={{ 'textDecoration': 'underline' }}>s</span>}}
  addnewparticle={addNewParticle}
/>
</div>
<div className="game-controllers">
  <button className="game-controllers__continue" id="game-controllers__continue"
onClick={checkParticleCombination}>Συνέχεια</button>
  <button className="game-controllers__reset" id="game-controllers__reset"
onClick={resetGame}>Επαναφορά</button>
</div>
</div>
)
};

```

```
export default ElemParticlesGame;
```

```
import React from 'react';
```

```
Particle Button Function
```

```

function ParticleButton({name, addnewparticle, load, index, id, displayName}) {
  const clickHandler = () => { //απο react περναμε απο τον "πατερα" στο "παιδι" μια funtion
    addnewparticle(name);
  };

  const keyHandler = (event) => { //πλήκτρο enter
    if (event.which === 13) {
      addnewparticle(name);
    }
  };
};

```



```

return (
  <div aria-label={"particle " + name + " with load " + load}
    tabIndex="0"
    role="button"
    key={index}
    className="bariums-mesons__particle"
    onKeyDown={event => keyHandler(event)}
    onClick={clickHandler}>
    <p id={id}>{displayName} <br/> {load} </p>
  </div>
)
}

```

```
export default ParticleButton;
```

Ορίζουμε το component Particle για κάθε σωματίδιο.

```

function Particle({ name, index, onClick, isClickable, classStyle, correctParticle,
hasCompletedStageTwo, triangleParticles }) {
  const [hasBeenChosenStageTwo, setHasBeenChosenStageTwo] = useState(false)

  const chooseToMatchWithDecay = () => {
    if (isClickable && correctParticle === name) {
      setHasBeenChosenStageTwo(!hasBeenChosenStageTwo)
      onClick(name)
    }
  }

  const mainClass = classStyle ? "particle" + classStyle : "particle" + " particle_" + index;
  const activeClass = hasBeenChosenStageTwo ? 'has-been-chosen-stage-two' : '';

  let newName;
  const hasOverline = name === "u_" || name === "d_" || name === "s_"
  newName = hasOverline ? <span style={{ 'textDecoration': 'overline' }}>{name.slice(0,
-1)}</span> : name;

  return (
    <div>
      <div className={` ${mainClass} ${activeClass}`} onClick={chooseToMatchWithDecay}>
        {newName}
      </div>
      {hasBeenChosenStageTwo && hasCompletedStageTwo &&
        <HadronTriangle triangleParticles={triangleParticles}/>}
    </div>
  )
}

```

```
export default Particle;
```

Function σχετικά με τις οδηγίες του παιχνιδιού.

```
import React from 'react';
```

```
function GameInstructions(props) {  
  return (  
    <div className="instructions" style={!props.isShowingInstructions && {display: 'none'}}>  
      Οδηγίες για το παιχνίδι  
      <button onClick={props.toggleInstructions}>X</button>  
    </div>  
  )  
}
```

```
export default GameInstructions;
```

ΑΛΛΗΛΕΠΙΔΡΑΣΕΙΣ ΣΤΟΙΧΕΙΩΔΩΝ ΣΩΜΑΤΙΔΙΩΝ

Για αρχή δίνονται στον μαθητή κάποια στοιχεία θεωρίας τα οποία θα τον εισάγουν στην έννοια της αλληλεπίδρασης των στοιχειωδών σωματιδίων μέσω των Διαγραμμάτων Feynman.

Συγκεκριμένα:

Ο πίνακας των Στοιχειωδών Σωματιδίων περιέχει τους δομικούς λίθους από τους οποίους αποτελείται η ύλη όλου του Σύμπαντος. Δεν μας λέει όμως, πως αλληλεπιδρούμε μεταξύ τους για να συγκροτήσουν αυτό το Σύμπαν. Είναι ακριβώς σαν να είχαμε τα γράμματα της αλφαβήτου αλλά όχι τις λέξεις και τους κανόνες που δημιουργούν προτάσεις με νόημα. Σίγουρα, στην καθημερινή ζωή μας, αντιλαμβανόμαστε ένα σχετικά μεγάλο αριθμό διαφορετικών δυνάμεων: ηλεκτρικές, βαρυτικές, μαγνητικές, τριβής κ.α. Πιστεύουμε όμως ότι όλο αυτό το πλήθος διαφορετικών αλληλεπιδράσεων που συναντάμε στη ύλη, μπορεί να καταταχθεί σε τέσσερις θεμελιώδεις αλληλεπιδράσεις, οι οποίες φαίνονται στον παρακάτω πίνακα:

Είδος αλληλεπίδρασης	Σχετικής ισχύς	Φορέας αλληλεπίδρασης	Παρουσία
Ισχυρή πυρηνική αλληλεπίδραση	~1	8 Γκλουόνια (άμαζα)	Ατομικούς πυρήνες
Ηλεκτρομαγνητική αλληλεπίδραση	~10 ⁻³	Φωτόνιο (άμαζο)	Ατομικές στιβάδες ηλεκτροτεχνία
Ασθενής πυρηνική αλληλεπίδραση	~10 ⁻⁵	Μποζόνια Z, W ⁺ , W ⁻	Ραδιενεργό διάσπαση
Βαρυτική αλληλεπίδραση	~10 ⁻³⁸	Βαρυτόνια (;)	Ουράνια σώματα

Η ηλεκτρομαγνητική δύναμη και η βαρύτητα ήταν γνωστές (τουλάχιστον σε κάποια μορφή τους), από τα αρχαία χρόνια και τις αντιλαμβανόμαστε στην καθημερινή μας ζωή. Η ισχυρή και η ασθενής πυρηνική δύναμη, παρ'όλο που δεν τις “αισθανόμαστε” άμεσα, παίζουν επίσης πολύ σοβαρό ρόλο στην ίδια μας την ύπαρξη. Ας δούμε όμως, μία μία τις τέσσερις αλληλεπιδράσεις.

ΒΑΡΥΤΙΚΗ ΑΛΛΗΛΕΠΙΔΡΑΣΗ

Αυτή η αλληλεπίδραση είναι υπεύθυνη για την συγκρότηση των γαλαξιών, για την περιφορά της Γης και των άλλων πλανητών γύρω από τον Ήλιο. Η επαναφορά μας στο έδαφος, όταν πηδάμε, οφείλεται στην βαρύτητα. Είναι η δύναμη που ασκείται μεταξύ όλων των σωμάτων και είναι πάντοτε έλκτική.

ΗΛΕΚΤΡΟΜΑΓΝΗΤΙΚΗ ΑΛΛΗΛΕΠΙΔΡΑΣΗ

Είναι υπεύθυνη για το “τίναγμα” που αισθανόμαστε όταν, περπατώντας πάνω σ'ένα χαλί, ακουμπάμε ένα σιδερένιο αντικείμενο. Είναι υπεύθυνη για την λειτουργία όλων των ηλεκτρικών μηχανών. Αλλά είναι ακόμα υπεύθυνη για την συγκρότηση των ατόμων: τα ηλεκτρόνια περιφέρονται γύρω από τον πυρήνα του ατόμου συγκρατούμενα από την ηλεκτρομαγνητική αλληλεπίδραση. Ασκείται μεταξύ όλων των σωμάτων που έχουν ηλεκτρικό φορτίο. Διακρίνουμε δύο ειδών φορτία: το θετικό και το αρνητικό. Τα ομόσημα φορτία (θετικο-θετικο ή αρνητικό-αρνητικό) απωθούνται ενώ τα ετερόσημα (θετικό-αρνητικό) έλκονται.

ΑΣΘΕΝΗΣ ΠΥΡΗΝΙΚΗ ΑΛΛΗΛΕΠΙΔΡΑΣΗ

Είναι η δεύτερη αλληλεπίδραση που δεν είναι άμεσα αντιληπτή. Παρ'όλα αυτά η σημασία της είναι μεγάλη. Απλώς αναφέρουμε εδώ, ως παράδειγμα, ότι μία από τις βασικές πυρηνικές αντιδράσεις που γίνονται στον ήλιο τον κάνουν να μας φωτίζει, οφείλεται ακριβώς σ'αυτήν την αλληλεπίδραση. Όσα σωματίδια αισθάνονται την ασθενή αλλά όχι την ισχυρή αλληλεπίδραση, τα ονομάζουμε λεπτόνια.

Η ΙΣΧΥΣ ΤΩΝ ΑΛΛΗΛΕΠΙΔΡΑΣΕΩΝ

Ας βάλουμε τώρα σε μία σειρά τις τέσσερις αλληλεπιδράσεις ανάλογα με την ισχύ τους. Ας πάρουμε την πιο ισχυρή από αυτές, την Ισχυρή Πυρηνική Αλληλεπίδραση (στην οποία δίνουμε αυθαίρετα ισχύ 1 στην δεύτερη στήλη του Πίνακα Αλληλεπιδράσεων) και ας συγκρίνουμε τις υπόλοιπες ως προς αυτήν. Η Ηλεκτρομαγνητική είναι περίπου 100 (για την ακρίβεια 137) φορές μικρότερη, ενώ η ασθενής πυρηνική δύναμη είναι 100.000 φορές πιο ασθενής από την ισχυρή. Για την Βαρυτική Αλληλεπίδραση, πραγματικά η ισχύς της είναι τραγικά μικρή. Γι'αυτό ακριβώς η βαρυτική αλληλεπίδραση δεν παίζει κανέναν ρόλο στα πειράματα που γίνονται με στοιχειώδη σωματίδια. Την “σκεπάζουν” κυριολεκτικά οι άλλες αλληλεπιδράσεις. Το αποτέλεσμα είναι να μην μπορούμε να έχουμε από αυτά τα πειράματα στοιχεία για την βαρυτική αλληλεπίδραση. Μόνο σε πειράματα που επεισέρχονται τεράστιες μάζες αστέρων (ή γαλαξιών) έχουμε την δυνατότητα να δούμε τα αποτελέσματα αυτής της αλληλεπίδρασης.Α

Στην συνέχεια ο μαθητής καλείται σε ένα διαδραστικό παιχνίδι με τα διαγράμματα Feynmann.

Τα διαγράμματα Feynman είναι ένας βασικός τρόπος απεικόνισης της αλληλεπίδρασης των στοιχειωδών σωματιδίων. Συγκεκριμένα ο οριζόντιος άξονας αναπαριστά το διάστημα και ο κάθετος τον χρόνο, οι ευθείες γραμμές αναπαριστούν φερμιόνια και οι κυματοειδείς μποζόνια.

Τα διαγράμματα Feynman απλοποίησαν την απεικόνιση των πολύπλοκων εξισώσεων που περιγράφουν την αλληλεπίδραση των σωματιδίων και τον μηχανισμό απορρόφησης και εκπομπής ακτινοβολιών. Στο συγκεκριμένο παιχνίδι εισάγουμε τον μαθητή στα διαγράμματα Feynman με τέτοιο τρόπο ώστε να εξοικειωθεί με την μορφή των αλληλεπιδράσεων στον άξονα του χώρου και του χρόνου.

Χρησιμοποιούμε τις μεθόδους της αριστερής περιστροφής, της δεξιάς περιστροφής και της αντιστροφής. Κατά την αντιστροφή ο μαθητής παρατηρεί την συμμετρία του διαγράμματος ως προς τον χρόνο και την μετατροπή του από σωματίδιο σε αντισωματίδιο και αντίστροφα.

Ο βασικός κώδικας του παιχνιδιού είναι ο εξής:

```
import React, { useState } from 'react';
import FeynmanInteraction from './feynmanInteraction';
import FeynmanRotation from './feynmanRotation';
```

```
const first_game_data_collection = [
  ["u", "arrow", "d", "W+"],
  ["W-", "u", "arrow", "d"],
  ["W-", "arrow", "u_", "d"],
  ["d_", "W-", "arrow", "u_"],
  ["d_", "arrow", "W+", "u_"],
  ["u", "d_", "arrow", "W+"],
];
```

```
const second_game_data_collection = [
  ["u", "arrow", "s", "W+"],
  ["W-", "u", "arrow", "s"],
  ["W-", "arrow", "u_", "s"],
  ["s_", "W-", "arrow", "u_"],
  ["s_", "arrow", "W+", "u_"],
  ["u", "s_", "arrow", "W+"],
];
```

```
const third_game_data_collection = [
  ["ve", "arrow", "e-", "W+"],
  ["W-", "ve", "arrow", "e-"],
  ["W-", "arrow", "ve_", "e-"],
  ["e+", "W-", "arrow", "ve_"],
  ["e+", "arrow", "W+", "ve_"],
  ["ve", "e+", "arrow", "W+"],
];
```

```

const fourth_game_data_collection = [
  ["νμ", "arrow", "μ-", "W+"],
  ["W-", "νμ", "arrow", "μ-"],
  ["W-", "arrow", "νμ_", "μ-"],
  ["μ+", "W-", "arrow", "νμ_"],
  ["μ+", "arrow", "W+", "νμ_"],
  ["νμ_", "μ+", "arrow", "W+"],
];

```

```

function FeynmanGame() {
  const [hasChosenInteraction, setHasChosenInteraction] = useState(false); //αν εχει διαλεξει
  reaction
  const [selected, setSelected] = useState({ react: "", result: "", dataCollectionSelected: []}); // τι
  reaction εχει διαλεξει

  const toggleOpenGame = () => { //αν εχει διαλεξει ανοιγει το game
    setHasChosenInteraction(!hasChosenInteraction)
  };

  const chooseInteraction = (react, result, index) => {
    toggleOpenGame();
    const dataCollectionSelected = index === 1 ? first_game_data_collection :
      index === 2 ? second_game_data_collection :
      index === 3 ? third_game_data_collection :
      index === 4 ? fourth_game_data_collection : null;
    setSelected({ react, result, dataCollectionSelected});
  };

  if (!hasChosenInteraction) { //αν δεν εχει διαλεξει τότε δειχνει τις οδηγιες
    return (
      <div>
        <header className="content-container__header">
          <h1>Feynman Διαγραμματα Παιχνιδι</h1>
          <h2>Από τους μετασχηματισμούς των Κουαρκ στις διασπάσεις των ανδρονίων</h2>
          <p>Παιξε με την ασθενη αλληλεπιδραση των λεπτονιων, των κουαρκ
            και του σωματιδιου W+ και, φυσικα, των αντισωματιδιων τους!
            Διαλεξε από τις παρακάτω κορυφές και δεσ πως αλληλεπιδρούν
            τα σωματιδια μεταξύ τους, στρέφοντας την κορυφη με τη φορά των
            των δεικτών του ρολογιού, με φορά αντίθετη των δεικτών του ρολογιού
            και παίρνοντας την κατοπτρική μορφή της κορυφής.
          </p>
        </header>
      </div>
    );
  }
}

```

```

<ul className="feynman-diagram--container">
  <li className="feynman-diagram">
    <FeynmanInteraction
      react={"u"}
      result={"d"}
      index={1}
      onClick={chooseInteraction}
    />
  </li>
  <li className="feynman-diagram">
    <FeynmanInteraction
      react={"u"}
      result={"s"}
      index={2}
      onClick={chooseInteraction}
    />
  </li>
  <li className="feynman-diagram">
    <FeynmanInteraction
      react={"v"}
      result={"e", <sup>-</sup>}
      index={3}
      onClick={chooseInteraction}
    />
  </li>
  <li className="feynman-diagram">
    <FeynmanInteraction
      react={"v"}
      result={"μ", <sup>-</sup>}
      index={4}
      onClick={chooseInteraction}
    />
  </li>
</ul>
</div>
} else {
  return (
    <div className="feynman-rotation-container">
      <FeynmanRotation toggleOpenGame={toggleOpenGame} initialReact={selected.react}
initialResult={selected.result} dataCollection={selected.dataCollectionSelected} />
    </div>
  )
}
}

```

```
export default FeynmanGame;
```

Έχουμε φτιάξει τα components FeynmanInteraction & Feynman Rotation τις οποίες εισάγουμε στον κώδικα και φαίνονται παρακάτω:

```
import React from 'react';
```

```
function FeynmanInteraction({ react, result, onClick, index }) {  
  const arrowHTML = <span>&#8594;</span>  
  
  const chooseInteraction = () => {  
    onClick(react, result, index)  
  }  
  
  return (  
    <div>  
      <input className="pointer" id={{react} + "_" + {result}} type="radio"  
onClick={chooseInteraction} />  
      <label for={{react} + "_" + {result}}  
        className="feynman-diagram__label pointer">  
        {react} {arrowHTML} {result} W<sup>+</sup></label>  
    </div>  
  )  
}
```

```
export default FeynmanInteraction;
```

```
import React, {useState} from 'react';
```

```
function FeynmanRotation({toggleOpenGame, dataCollection}) {  
  const arrowHTML = <span>&#8594;</span>
```

//παίρνει το data collection κι εμφανίζει την θέση των σωματιδίων κάθε στιγμή. 3 θέσεις συγκριμένες στις οποίες βρίσκεται κάθε φορά το σωματίδιο.

```
  const changeDisplayedDataCollection = (dataCollection) => dataCollection.map(position =>  
position.map(particle => {  
  switch (particle) {  
    case 'u_':  
      return <span style={{ 'textDecoration': 'underline' }}>u</span>  
    case 'u':  
      return <span>u</span>  
    case 'd':  
      return <span>d</span>  
    case 'd_':  
      return <span style={{ 'textDecoration': 'underline' }}>d</span>  
    case 's':  
      return <span>s</span>  
    case 's_':
```

```

return <span style={{ 'textDecoration': 'overline' }}>s</span>
  case 've':
    return <span>v<sub>e</sub></span>
  case 'e-':
    return <span>e<sup>-</sup></span>
  case 'e+':
    return <span>e<sup>+</sup></span>
  case 've_':
    return <span style={{ 'textDecoration': 'overline' }}>v<sub>e</sub></span>
  case 'μ+':
    return <span>μ<sup>+</sup></span>
  case 'μ-':
    return <span>μ<sup>-</sup></span>
  case 'νμ':
    return <span>ν<sub>μ</sub></span>
  case 'νμ_':
    return <span style={{ 'textDecoration': 'overline' }}>ν<sub>μ</sub></span>
  case 'W+':
    return <span>W<sup>+</sup></span>
  case 'W-':
    return <span>W<sup>-</sup></span>
  case 'arrow':
    return "arrow"
  }
}))

```

```

const [currentIndex, setCurrentIndex] = useState(0);
const [currentRelationship, setCurrentRelationship] = //ορίζει ποιό είναι στα αριστερά
useState(changeDisplayedDataCollection(dataCollection)[0]); //αλλάζει θέση στο data collection
που καλείται
const [isRotationEnabled, setIsRotationEnabled] = useState(false);
const [reversedDataCollection, setReversedDataCollection] =
useState(changeDisplayedDataCollection(dataCollection));

const rotateRight = () => {
  let newIndex = currentIndex;
  if (dataCollection.length-1 > currentIndex) {
    setCurrentIndex(newIndex+1);
    isRotationEnabled ?
setCurrentRelationship(changeDisplayedDataCollection(dataCollection)[newIndex+1])
: setCurrentRelationship(reversedDataCollection[newIndex+1]);
  } else {
    setCurrentIndex(0);
    setCurrentRelationship(reversedDataCollection[0]);
  }
};

```



```

const rotateLeft = () => {
  let newIndex = currentIndex;
  if ( currentIndex === 0 ) {
    setCurrentIndex(dataCollection.length-1);
    !isRotationEnabled ?
setCurrentRelationship(changeDisplayedDataCollection(dataCollection)[dataCollection.length-
1])
      : setCurrentRelationship(reversedDataCollection[dataCollection.length-1]);
  } else {
    setCurrentIndex(newIndex-1);
    setCurrentRelationship(reversedDataCollection[newIndex-1]);
  }
};

const rotateOpposite = () => {
  // const newRotationStatus = !isRotationEnabled;
  //
  // const newReversedDataCollection = reversedDataCollection.map((relationship, index) => {
  //   let newRelationship = relationship.slice();
  //   newRelationship[0] = relationship[3];
  //   newRelationship[1] = relationship[2];
  //   newRelationship[2] = relationship[1];
  //   newRelationship[3] = relationship[0];
  //   return newRelationship;
  // });
  //
  // setReversedDataCollection(newReversedDataCollection);
  // setCurrentRelationship(newReversedDataCollection[currentIndex]);
  // setIsRotationEnabled(newRotationStatus);

  let newIndex = (dataCollection.length-1) >= (currentIndex+3) ? (currentIndex+3) :
((currentIndex+3) - (dataCollection.length))
  setCurrentIndex(newIndex);
  isRotationEnabled ?
setCurrentRelationship(changeDisplayedDataCollection(dataCollection)[newIndex])
  : setCurrentRelationship(reversedDataCollection[newIndex]);
};

return (
  <div className="triangle-container">
    { currentRelationship.indexOf('arrow') === 1 ?
      <div className="one_left_triangle">
        <div className="center"></div>
        <div className="left-bar"></div>
        <div className='particle left-particle'>
          {currentRelationship[0]}
        </div>

```

```

<div className="left-top-bar"></div>
  <div className='particle left-top-particle'>
    {currentRelationship[2]}
  </div>
  <div className="left-bottom-bar"></div>
  <div className='particle left-bottom-particle'>
    {currentRelationship[3]}
  </div>
</div> :
<div className="two_left_triangle">
  <div className="center"></div>
  <div className="right-bar"></div>
  <div className='particle right-particle'>
    {currentRelationship[3]}
  </div>
  <div className="right-bottom-bar"></div>
  <div className='particle right-bottom-particle'>
    {currentRelationship[0]}
  </div>
  <div className="right-top-bar"></div>
  <div className='particle right-top-particle'>
    {currentRelationship[1]}
  </div>
</div>
}
<div className="feynman-triangle__controls">
  <div className="feynman-triangle__rotations">
    <button className="main-menu__option space main-menu__option--first"
onClick={rotateLeft}>Αριστερή Περιστροφή</button>
    <button className="main-menu__option space main-menu__option--first"
onClick={rotateRight}>Δεξιά Περιστροφή</button>
  </div>
  <div className="feynman-triangle__reverse">
    <button className="main-menu__option space main-menu__option--second"
onClick={rotateOpposite}>Αντιστροφή</button>
  </div>
  <div className="feynman-triangle__back">
    <button className="main-menu__option space" onClick={toggleOpenGame}>Πίσω
στην Θεωρία</button>
  </div>
</div>
</div>
)
}

```

```
export default FeynmanRotation;
```

```
import React from 'react';

function FeynmanDiagram(props) {
  return (
    <div>
      Diagram
    </div>)
}

export default FeynmanDiagram;
```