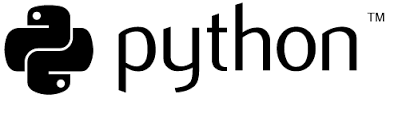
**ΟΜΑΔΑ 06**



**Πρόλογος**

**Σοφοκλής Παρούτης ΑΜ: 1050658**

**Μάριος Μιλτιάδους ΑΜ:1021041**

**Μάριος Βασιλείου ΑΜ:1021043**

**Νίκος Χαραλάμπους ΑΜ:1021047**

**1/31/2016**

**Η σωστή, λοιπόν,  τοποθέτηση του ζητήματός μας είναι να πούμε ότι η πρόοδος της επιστήμης και της τεχνολογίας μπορεί να καλυτερέψει τον άνθρωπο, αφού δημιουργεί τις προϋποθέσεις για την πνευματική του απογείωση. Ας το πάρουμε μια για πάντα απόφαση: αφού μπήκαμε στο δρόμο των επιστημών και τεχνικών κατακτήσεων. Στο ακαδημαικό έτος 2015-16 μας έχει ανατεθεί στο μάθημα Εισαγωσή στους ηλεκτρονικούς υπολογιστές ένα προτζεκτ που αφορά στην καινοτομία και συμφιλίωση με μια πλατφόρμα ανοικτού κώδικα που ονομάζουμε Kivy και η οποία επιτρέπει στον χρήστη να αναπτύσσει εφαρμογές που εμφανίζουν καινοτόμα interfaces τύπου multi-touch. στην πλατφόρμα Kivy.**

**Ρόλος Μελών:**

**Στην αρχή αφού πληροφορηθήκαμε για την εργασία μας, όλη η ομάδα κάναμε μια συνάντηση και αναθέσαμε στο κάθε μέλος από ένα μέρος του προτζεκτ- προγράμματος. Αφού οργανωθήκαμε και καθένας από ‘μάς έκανε την σχετική δουλεία που του ανατέθηκε, βρεθήκαμε μετά από λίγες μέρες και συνδέσαμε όλα τα κομμάτια του πρότζεκτ έτσι ώστε να βγεί το επιθημιτό αποτελέσμα. Στην πορεία όμως εντοπίστηκαν κάποια προβλήματα τα οποία μετά την συνεργασία μας βοηθήσαμε ο ένας τον άλλο και τα ξεπεράσαμε.**

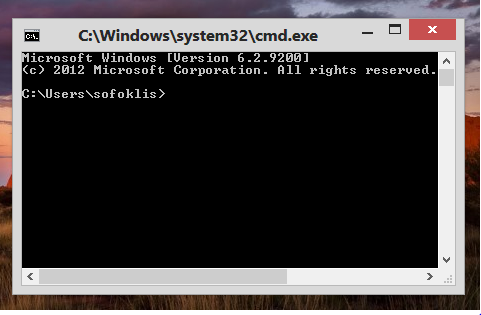
**Ο Μάριος Βασιλείου έχει πάρει το κομμάτι της έρευνας το οποίο αφορούσε στην πληροφόρισή σας τί είναι το Kivy. Παράλληλα ο ρόλος του Νίκου Χαραλάμπους ήταν να βρεί γιατί σχεδιάστηκε το Kivy και ποίος είναι ο στόχος του. Επιπρόσθετα ο Μάριος Μιλτιάδους ασχολίθηκε με το τί μπορούμε να επιτύχουμε με την πλατφόρμα του Kivy. Εντέλη ο Σοφοκλής Παρούτης αναζήτησε για την λειτουργία της πλατφόρμας με σχετικά βίντεο και εικόνες της παρουσίασης αυτής.**

**Εν κατακλείδι χωριστήκαμε σε δύο υποομάδες με σκοπό η μια υποομάδα (Μάριος Βας. και Νίκος) ασχολήθηκε με την συλλογή πληροφοριών με αποτέλεσμα την δημιουργεία της σημερινής παρουσίασης (power point), ενώ η δεύτερη υποομάδα (Σοφοκλής και Μάριος) με την εγκατάσταση και έλεγχον τον προγραμμάτων.**

**Μέθοδος επίλυσης του προβλήματος**

**Για να ενημερωθούμε σχετικά με την συγκεκριμένη πλατφόρμα επισκεφθείκαμε καταρχήν την διεύθυνση <https://kivy.org/#home>.**

**Το μόνο που χρειάζεται στην κανονική εγκατάσταση είναι η Python 2.7 με τις αναβαθμίσεις της και το pyGame.**

**Στην συγκεκριμένη σελίδα βρείκαμε πληροφορίες για την εγκατάσταση του Kivy για τις εντολές που διαθέτει καθώς και παραδείγματα κώδικα που μπορέσαμε να τρέξουμε για να αντιληφθούμε την λειτουργία του. Πρώτα απ’όλα για να τρέξει το πρόγραμμα Kivy έπρεπε να εγκαταστήσουμε το pip, το pygame και το Kivy. Για την εγκατάσταση ανοίξαμε το ποιό κάτω παράθυρο .**

**Στο παράθυρο αυτό γράφουμε με σειρά τις ποιό κατω εντολές :**

1. **pip install --upgrade pip wheel setuptools**
2. **pip install docutils pygments pypiwin32 kivy.deps.sdl2 kivy.deps.glew kivy.deps.gstreamer --extra-index-url** [**https://kivy.org/downloads/packages/simple/**](https://kivy.org/downloads/packages/simple/)
3. **pip install kivy**

**Το μόνο που χρειάζεται κανονική εγκατάσταση είναι η Python 2.7 με τις αναβαθμίσεις της και το pyGame.**

**Πηγές**

**Για την εγκατάσταση της python:**

[**https://www.python.org/downloads/**](https://www.python.org/downloads/)

**Για την εγκατάσταση τoυ pygame:**

[**http://www.pygame.org/download.shtml**](http://www.pygame.org/download.shtml)

**Για την εγκατάσταση τoυ pygame και kivy:**

[**http://www.lfd.uci.edu/~gohlke/pythonlibs/**](http://www.lfd.uci.edu/~gohlke/pythonlibs/)



**Πληροφορίες σχετικά με το kivy:**

[**https://kivy.org/#home**](https://kivy.org/#home)

**Kώδικας:**

**from kivy.app import App**

**from kivy.uix.widget import Widget**

**from kivy.properties import NumericProperty, ReferenceListProperty,\**

**ObjectProperty**

**from kivy.vector import Vector**

**from kivy.clock import Clock**

**class PongPaddle(Widget):**

**score = NumericProperty(0)**

**def bounce\_ball(self, ball):**

**if self.collide\_widget(ball):**

**vx, vy = ball.velocity**

**offset = (ball.center\_y - self.center\_y) / (self.height / 2)**

**bounced = Vector(-1 \* vx, vy)**

**vel = bounced \* 1.1**

**ball.velocity = vel.x, vel.y + offset**

**class PongBall(Widget):**

**velocity\_x = NumericProperty(0)**

**velocity\_y = NumericProperty(0)**

**velocity = ReferenceListProperty(velocity\_x, velocity\_y)**

**def move(self):**

**self.pos = Vector(\*self.velocity) + self.pos**

**class PongGame(Widget):**

**ball = ObjectProperty(None)**

**player1 = ObjectProperty(None)**

**player2 = ObjectProperty(None)**

**def serve\_ball(self, vel=(4, 0)):**

**self.ball.center = self.center**

**self.ball.velocity = vel**

**def update(self, dt):**

**self.ball.move()**

**#bounce of paddles**

**self.player1.bounce\_ball(self.ball)**

**self.player2.bounce\_ball(self.ball)**

**#bounce ball off bottom or top**

**if (self.ball.y < self.y) or (self.ball.top > self.top):**

**self.ball.velocity\_y \*= -1**

**#went of to a side to score point?**

**if self.ball.x < self.x:**

**self.player2.score += 1**

**self.serve\_ball(vel=(4, 0))**

**if self.ball.x > self.width:**

**self.player1.score += 1**

**self.serve\_ball(vel=(-4, 0))**

**def on\_touch\_move(self, touch):**

**if touch.x < self.width / 3:**

**self.player1.center\_y = touch.y**

**if touch.x > self.width - self.width / 3:**

**self.player2.center\_y = touch.y**

**class PongApp(App):**

**def build(self):**

**game = PongGame()**

**game.serve\_ball()**

**Clock.schedule\_interval(game.update, 1.0 / 60.0)**

**return game**

**if \_\_name\_\_ == '\_\_main\_\_':**

**PongApp().run()**