

Визуализация на мозъчни вълни получени от слушалки MindSet

Георги Кръстев, Ваня Динева

Visualization of brain waves generated by MindSet headset: This article presents software for visualization of brain waves generated by MindSet headset. There is a description of the possibilities for real-time visualization, when are data loaded and actions when data recording.

Key words: Feedback Methods, Visualization, Brain Waves, MindSet.

ВЪВЕДЕНИЕ

Благодарение на компютърните технологии и наличието на разнообразни сензори, човек може не само да визуализира който и да е вътрешен процес (мускулно напрежение, кръвоток, мозъчна активност и др.), но и да се научи да го контролира и управлява чрез мултимедийни презентации на монитор, наблюдавайки го без мишка, клавиатура или джойстик. В метафоричен смисъл може да се каже, че се касае за мозъчен фитнес. Провеждането на тренировките до голяма степен приличат на тези при боди-билдинга. Но вместо да трупа мускули, човек усъвършенства своето внимание, концентрация, стабилизира настроението си, бързо се освобождава от нервнопсихично напрежение и релаксира, подобрява функциите на паметта и ускорява процеса на вземане на оптимално решение. Всичко това е възможно с фийдбек методите на приложната неврофизиология, в чиято основа е повлияването на психичното състояние, променящо физиологичните телесни процеси и обратно. Те се използват не само за тренинг, но и за диагностика и терапия на различни болестни разстройства - хиперактивност и дефицит на вниманието, шизофрения, алцхаймер, депресия, алкохолна и други видове зависимости, тревожност, аутизъм, епилепсия и др. В този смисъл когато човек наблюдава в реално време на монитор своята мозъчна биоелектрична активност може сам да повлияе на конкретните параметри в желана от него посока.

Настоящата статия представя разработено програмно осигуряване за визуализация на мозъчни вълни получени от слушалки MindSet [1, 2, 7].

Разработката е на Visual Studio 2012 на C# със софтуерна рамка .NET 3.5.

ОПИСАНИЕ НА ПРОГРАМНИТЕ МОДУЛИ

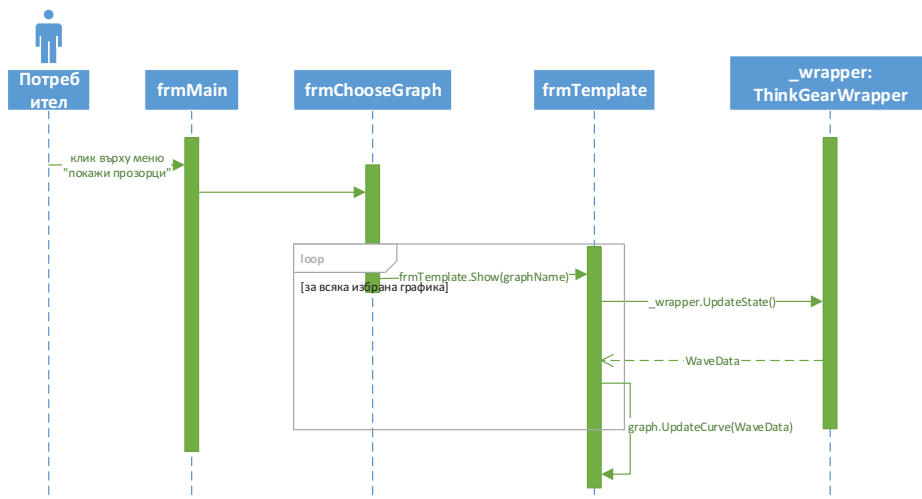
За конкретното програмно осигуряване са използвани модули и библиотеки от Интернет с отворен код:

- **ZedGraph** – Zed Graph се предоставя под LGPLv2 лиценз. Това е библиотека, с помощта на която се постига визуализацията на отделните мозъчни вълни. ZedGraph е изключително податлива на персонализиране. Тя се предоставя със сравнително подробна документация. Предоставя на потребителите си както клас, с който да работят, така и елемент, който може да се постави във всяка една Windows форма (<http://sourceforge.net/projects/zedgraph/>);
- **Loader** – клас, с помощта на който записани данни могат да се заредят в програмата с цел визуализация;
- **DataEntry** – спомагателен клас, служещ за съхранение на данни за всяка една от мозъчните вълни, разпознати от слушалките, за единица време;
- **DataContainer** – този клас съдържа списък от DataEntry подредени по време и предоставя функционалността за запис на данни във файл;
- **frmTemplate** – този клас предоставя прозорец, служещ за шаблон. Тук се намира инстанцията на ZedGraph. Този прозорец се създава повече от веднъж;
- **frmChooseGraph** – това представлява диалогов прозорец, с помощта на който потребителят може да избере кои мозъчни вълни да визуализира.

Визуализация в реално време

Нужно е да се отбележи, че за да е възможна визуализацията в реално време програмата трябва да е вече свързана към слушалките. Ако програмата не е свързана към тях при натискане и опит за избиране на графика ще се изведе съобщение за грешка.

Модулите, имащи връзка по между си в този случай са: frmMain, frmChooseGraph, frmTemplate и _wrapper (фиг. 1).



Фиг. 1. Визуализация на мозъчни вълни в реално време

След като потребителят кликне „Прозорци->Избиране на графики“ се инициализира инстанция на frmChooseGraph. От там той избира желаните графики за визуализация и натиска бутонът ОК върху формата, имената на избраните графики се съхраняват в публично достъпният речник cbList, като за ключ на този речник се използва името на вълната, а стойността е самият checkbox елемент. При затварянето на инстанцията на frmChooseGraph във frmMain се извършват различни проверки за да се изключи добавянето на вече избрани графики отново и др. В резултат на това се създава подреден списък като всяка двойка ключ/стойност на този списък има за ключ името на желаната графика и стойност инстанция на frmTemplate. В момента на добавяне на записите в списъка всяка инстанция на frmTemplate бива инициализирана и бива извикан метода frmTemplate.Start() на тази инстанция.

Методът frmTemplate.Start() има грижата да създаде нишка, да и зададе работа във фонов режим и да я стартира. Животът на нишката продължава докато инстанцията на frmTemplate не бъде затворена. Докато е отворена инстанцията нишката опреснява състоянието на _wrapper и взема нужните данни спрямо типа вълна за който е била създадена. След като има данните, те биват добавени и начертани на графиката.

Например, ако потребителят желае да изчертава графики на всички видове вълни: Алфа Ниски, Алфа Високи, Бета Ниски, Бета Високи, Гамма Ниски, Гамма Високи, Тета, Медитация и Концентрация, основният прозорец ще има 10 на брой отворени прозореца в себе си, програмата ще има 12 индивидуални нишки. 10

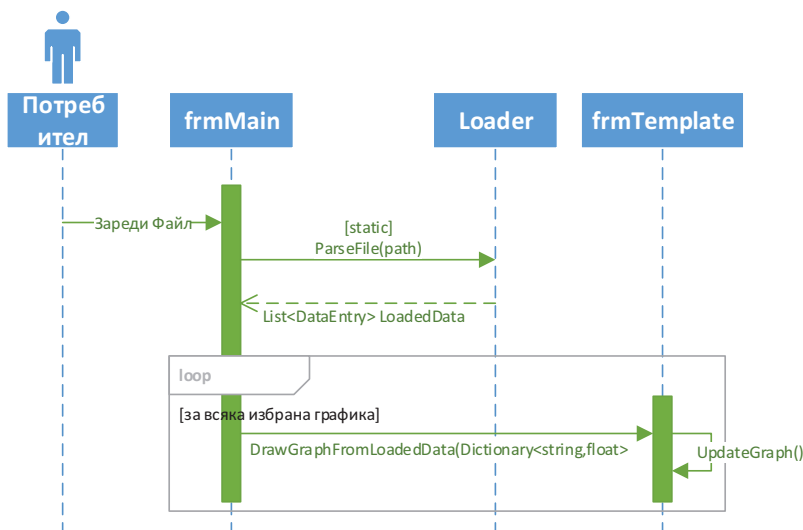
за всяка една вълна, 1 за управлението на основният прозорец и една за поддръжка на информацията за нивото на сигнала. Информацията за всички 10 вълни ще идва от една и създа инстанция на класа ThinkGearWrapper (_wrapper).

Визуализация при заредени данни

За разлика от случая когато визуализираме данни в реално време, при визуализацията на заредени данни не трябва да се свързваме със слушалките.

От фиг. 2 се вижда, че в този случай модулите, които си взаимодействат са: frmMain, frmTemplate и класът Loader.

За да започне визуализацията потребителят трябва първо да зареди предварително записани данни във програмата. Ако погледнем декларираните променливи ще видим списъкът LoadedData, който е предназначен да съдържа данни от тип DataEntry. При клик на „Зареди Запис“ от менюто се отваря диалогов прозорец подканващ потребителят да избере файл. След това пътят към този файл бива предаден на статичният метод ParseFile на класа Loader, а върнатата стойност се присвоява на LoadedData. В този метод файла бива прочетен в поток. Файлът се чете ред по ред като първият прочетен ред бива проверяван за специален словоред. При положение, че този словоред не е спазен файлът се счита за невалиден и се извежда съобщение за грешка. Ако валидизацията премине успешно файлът продължава да се чете ред по ред докато не се достигне до края му. Всеки ред от файла се записва в локална променлива _Container, която е от същия тип като LoadedData спомената по-горе, като тя бива върната след края на четенето. Така вече програмата разполага с данни, които може да визуализира.



Фиг. 2. Визуализация при заредени данни

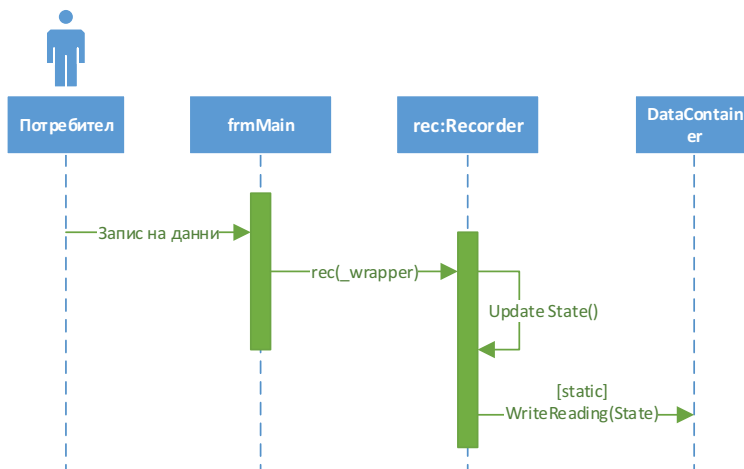
Визуализирането на данните става по същия начин както при визуализирането в реално време. Разликата е в това, че инстанция на ThinkGearWrapper не се налага да се създава. Когато се създават инстанциите на frmTemplate вместо програмата да се обръща към методът frmTemplate.Start(), тук тя се обръща към frmTemplate.DrawGraphFromLoadedData(Data). Аргументът Data представлява

речник от string като ключ и float като стойност. Ключът представлява времето след старта на записа, а стойността е показанието в този момент.

Запис на данни във файл (Export)

Друга функционалност предоставяща програмата е запис на данни във файл. На фиг. 3 е представена диаграмата на последователностите нужна за изпълнението на тази опция. Както се вижда от диаграмата тук модулите комуникиращи един с друг са: frmMain, _wrapper, класът Recorder и класът DataContainer.

След като потребителят избере „Запис->Започни запис“ от менюто на основният прозорец, първо се прави проверка за наличие на незаписани данни (останали в паметта) от предишна запис сесия. При наличие на такива данни потребителят се запитва какво да стане с тях, като опциите са две - да се запишат или да се изтрият. При избор да се запишат потребителят е приканен да посочи път където да се запише файла, след което стартира нова сесия. При отговор да се изтрият, данните се изтриват и започва нова сесия. Възможно е потребителят да се съгласи да запише данните, но да се откаже, когато му е предоставена възможността да посочи пътят за запис на файла, тогава се извежда съобщение, че нова сесия не е започнала.



Фиг. 3. Последователност на действията при запис на данни

Сесията за запис започва с инициализирането на обект от класа Recorder. При създаването на този обект като параметър на конструктора се подава инстанция на ThinkGearWrapper (_wrapper). След като обектът (rec) вече има копие на _wrapper се създава нишка, която работи във фонов режим. Нишката има за цел да обновява състоянието на _wrapper през определен интервал от време и да записва сметите данни за всички вълни във статичния клас DataContainer.

По време на сесията за запис се извежда нотификация в главният прозорец посредством статус лентата в долният край на прозореца.

ЗАКЛЮЧЕНИЕ

В доклада е разгледано програмното осигуряване за визуализация на мозъчни вълни получени от слушалки MindSet. Представени са възможностите за визуализация в реално време, при заредени данни и последователността на действията при запис на данни. Вариациите на честотите на мозъчните вълни могат да бъдат визуализирани както с цифри и графики, така и с компютърни игри, при които трениращият или губи, или придобива повече контрол в играта. При това невротренингът се явява един добър метод за обучение и премодулиране на мозъчните вълни като в практиката за постигане на по-голяма ефективност много често се комбинира с психотерапия.

ЛИТЕРАТУРА

- [1] Кръстев, Г. Технически решения за човеко-машинен интерфейс чрез анализ на активността на мозъка, Научни трудове, РУ&СУ 14, т. 53, с. 3.2, стр. 54-58, 2014.
- [2] Кръстев, Г. Комуникационна среда на човеко-машинен интерфейс със слушалки MindSet, Научни трудове, РУ&СУ 14, т. 53, с. 3.2, стр. 87-91, 2014.
- [3] Karray, F. et al. Human-Computer Interaction: Overview on State of the Art. International Journal On Smart Sensing and Intelligent Systems, Vol. 1, No.1, March 2008.
- [4] Preece, J., Y. Rogers, H. Sharp (2002). Interaction Design: Beyond Human-Computer Interaction. New York, NY: John Wiley & Sons.
- [5] Preece, J., Y. Rogers, H. Sharp, D. Benyon, S. Holland, T. Carey (2004). Human-Computer Interaction. Wokingham (GB): Addison-Wesley.
- [6] <http://brain-amigo.com/>
- [7] <http://www.neurosky.com/>

За контакти:

проф. д.н. Георги Кръстев, Катедра "Компютърни системи и технологии", Русенски университет "Ангел Кънчев", тел.: 082-888 672, e-mail: gkrastev@ecs.uni-ruse.bg

гл. ас. д-р Ваня Динева, Катедра "Педагогика, психология, история", Русенски университет "Ангел Кънчев", e-mail: vdineva@uni-ruse.bg

Докладът е рецензиран.