Докато не са излезли екипите за проекта, си позволих да драсна една идея върху, която да работим. Все пак е по - добре да има нещо от колкото да тръгнем от бял лист. Всичко написано по - долу е моето виждане за нещата, естествено какво ще се прави е колективно решение и може и нищо от това да не се случи, но така и така нищо друго не може да се работи до излизането на екипите и реших да запиша някои мои идеи.

**Работна идея:**

Да създадем 2D игра „Soft Trainer” (името не е от голямо значение) със следния сценарий:

Действието да се развива на правоъгълно игрално поле, в което долната част ще е играча (трейнъра), който може да се движи из цялото поле, а в горната ще са обектите (неуките студенти), които също могат да се движат. Целта на играта е трейнъра да „елиминира” всички неуки студенти, като стреля по тях с куршуми: „C#”, „JS”, “PHP”, “Java”. Куршумите ще са краен брой и след като се изчерпят и все още има останали неуки студенти, трейнъра ще е принуден да ги прасне с клавиатурата по главата за да им налее знания. Този метод за обучение ще изисква неукия студент да е с гръб към трейнъра. Трейнъра няма да е неуязвим, неуките студенти ще бълват тъпи въпроси „DQ” (възможно е “DQ” да са самонасочващи към трейнъра и да трябва и те да се неутрализират), които попадайки на трейнъра ще вземат от неговото търпение и след като той изгуби цялото си търпение той напуска SoftUni и неуките студенти си остават неуки.

Възможни начини за изпълнение на някои от нещата (до толкова колкото мога да се сетя):

Игралното поле да се направи като структура с координати (не че разбирам защо, но така беше направена играта Falling Rocks ☺).

Анимацията на обектите в играта да стане, чрез използване на нишки.

**Трейнъра** да се имплементира като class Trainer:

**Полета:**

* private int x (координати по х);
* private int у (координати по у);
* private int patience;
* private Object image;
* private string (или Object) aimed (с какви куршуми е зареден ☺). Куршумите може да са в масив private static string (или Object) bullets;
* private string LooksAt ще може да приема стойности „left”, “right” “up” и “down”. (ще е необходимо за определяне първоначалната посока на куршума);
* private string playerName; и съответния конструктор PlayerName {get; set;}

**Методи:**

* Методи за движение, които ще се извикват при натиснати W, S, A или D (да се управлява с лявата ръка, дясната да е свободна за стрелва от мишката) за съответната посока.
* Метод за смяна на въоръжението на трейнъра (може да стане с един switch да приема натиснатата цифра от клавиатурата(1, 2, 3, 4, 5, 6…) да се зарежда съответния елемент от масива и да се присвоява на aimed, след което да се променя картинката на трейнъра със съответното оръжие. (трябва да се разберем, ако има време може картинките на оръжията да са за C# - логото на Visual Studio, за Java - логото на Eclipse, за JavaScript - логото на Web Storm и за PHP - логото на PHP Storm). Възможен вариант е и превърта куршумите, чрез натискане на десния бутон на мишката.
* Метод за изстрелване на куршум при натискане на мишката.

**Куршума** да е class Bullet:

**Полета на** Bullet:

* + private int x (координати по х);
  + private int у (координати по у);
  + private Object image;( == aimed)
  + public Bullet (конструктор, който да създава куршума);

**Методи на** Bullet:

* + Метод, който да движи куршума в изстреляната посока;
* Метод за „обучение чрез клавиатура” ☺ тук виждам един основен проблем как да се определи, че местоположението на трейнъра е зад студента, за да е възможно да го фрасне със клавиатурата точно отзад. Аз си мисля, че ако за да е зад студента и двамата трябва да гледат в една посока и трейнъра да е на разстояние с едно по-малко от координатите на неукия студент.

**Неуки студенти** да се създаде като class DumpStudent с полета:

**Полета:**

* private int x (координати по х);
* private int у (координати по у);
* private Object image;
* private int stupidity; (колко е глупав студента)
* private string LooksAt ще може да приема стойности „left”, “right” “up” и “down”. (ще е необходимо за удара с клавиатурата и за изплюването на тъп въпрос „DQ”);
* private const int ThrowDumpQuestion - константа, която ще пази през колко време ще се

**Методи:**

* Методи за движение (може да се направи метод, който да приема произволна faceOrientation и да увеличава(намалява) с единица х(у) координатите на студента, в зависимост от това на коя страна е обърнат).
* Метод за бълване на тъп въпрос от страна на студента.

**Тъпия въпрос** да е class DumpQuestion:

Полета на DumpQuestion:

* + private int x (координати по х);
  + private int у (координати по у);
  + private Object image;
  + public DumpQuestion (конструктор, който да създава въпроса);

**Методи на** DumpQuestion:

* + public MoveToMakeAngry - този метод ще придвижва тъпия въпрос към Трейнъра. Възможен вариант това да се случи е като се вземат координатите х и у на Трейнъра и координатите х и у на DumpQuestion, след което се сравнят и ако разликата е 0, не се променя координата, ако е отрицателна координатата се увеличава с 1, ако е положителна се изважда 1. Може и една променлива да следи дали въпроса е ударил трейнъра и една променлива да следи дали въпроса е ударил куршум на трейнъра след като вземе координатите на куршума.

**Научени студенти** да се създаде като class CleverStudent, който да се получи чрез молиморфизъм от DumpStudent, след като stupidity на DumpStudent стане по - малко или равно на нула. Може вече умния студент или да драсва на някъде докато стигне извън екрана, или да изчезва, или да помага на Трейнъра, като продължава да се движи из игралното полето, но „яде” глупавите въпроси.

**Система за левелинг**:

* Метод, който да дава точки на трейнъра в зависимост от това колко куршуми са му останали, кое ниво е преминал, колко студенти е просветил или по някакъв друг критерии.
* Метод, който да зарежда нива;
* Метод, който да задава сложност на нивото (наличието на n на брой тъпи ученика и времетраене, през което се произвеждат въпросите в същия метод може да заложим кога да се „превърта” играта (да е преминато последното ниво))
* Поле в структурата private int score, в което да се пази натрупаните точки на играча.

**Забележки и възможни подобрения при наличия на време:**

1. Първоначално да направим куха структура на играта от класове и полета в тях (може структурата да се пише на компютъра на един колега, а всички останали да гледат чрез Skype и да дават акъли и да се обсъжда самата структура с цел качествен код)
2. След това да сътворим някакъв работещ вариант на играта… примерно с 2 куршума, само с един трейнър и до 3 тъпи студента на едно игрално поле. И след това да добавяме допълнителни функционалности.
3. Може да се създаде class SoftUnits ☺, който да е „родител” на class Trainer, class DumpStudent и class CleverStudent (чак тука се сетих за това подобрение в кода и затова е написано тук, но структурата може да се изгради по този начин).
4. Може да се създаде class SoftUniWepons, който да е „родител” на class Bullet и class DumpQuestion. (чак тука се сетих за това подобрение в кода и затова е написано тук, но структурата може да се изгради по този начин)
5. Може да направим да се избира в началото из между всички трейнъри в Софтуни.
6. Може да се създаде и допълнителна функционалност на трейнъра (SoftUniConf), която да премахва всички тъпи въпроси и тъпи студенти в 2 координата радиус около трейнъра.
7. Може тъпите въпроси да има вид и да трябва да се неутрализират със същия вид куршум (това трябва да го решим още в началото при изграждане на структурата защото последващата му имплементация може да доведе до мазало).
8. Може да се създаде класация за High Score.
9. Трябва да предвидим и време за тестване на геймплейя. Може играта да се окаже прекалено лесна, или прекалено сложна, или да засича, или да се хващат някакви тактики, които да я направят скучна ☺.
10. Може играта да се направи 2.5D ☺ тоест да има и координати по z, които да се изразяват на екрана с по - тъмен цвят на фона или по друг начин. Което според мен няма да е кой знае какъв допълнителен ефект, а ще доведе до създаване възможност за доста бъгове.

Визуализация ☺:

