# Key algorithms

# Weapon Choice

class WeaponChoice() //class of a menu where you can choose a weapon

STATEMENT DatabaseImportstmt //opens a statement, which allows to work with the database

SELECT DatabaseImport //selects needed data from database

resultSet WeaponChoiceCondition //condition, stored in the database, which enables the weaponPurchase menu, if TRUE. Initially is FALSE

end SELECT

end STATEMENT

integer WeaponchoiceCount = 0 //The counter for the class. Increases each time when the weapon is chosen

String Weapon1orWeapon2string //The string will get the value of the chosen weapon between 2 options, so it can be updated to database

String Weapon3orWeapon4string

...

String Weapon7orWeapon8string

button Weapon1Choice //Allows the user to choose the weapon by pressing the button

button Weapon2Choice

...

button Weapon7Choice

button Weapon8Choice

button QuitMenu //button that allows to come back to buy menu. As after pressing this button no actions can be taken, it used for   
 //updating the data in database

QuitMenu.disable() //the quit button should be initially disabled, as some actions needed to be taken before quitting

if Weapon1 = action.pressed then //if the Weapon1 button is pressed, therefore the user preferred Weapon1 over Weapon2

Weapon1Choice.disable() //disables the pressed button, and the button for the opposite weapon

Weapon2Choice.disable()

String Weapon1orWeapon2string = "Weapon 1" //The string gets the value of the chosen weapon, so it can be updated to database

WeaponchoiceCount = WeaponchoiceCount + 1 //the counter increases each time when the choice between two weapons is made

end if

if Weapon2 = action.pressed then //if the Weapon2 button is pressed, therefore the user preferred Weapon2 over Weapon1

Weapon1Choice.disable()

Weapon2Choice.disable() //disables the pressed button, and the button for the opposite weapon

String Weapon1orWeapon2string = "Weapon 2" //The string gets the value of the chosen weapon, so it can be updated to database

WeaponchoiceCount = WeaponchoiceCount + 1 //the counter increases each time when the choice between two weapons is made

end if

...

if Weapon8 = action.pressed then //if the Weapon8 button is pressed, therefore the user preferred Weapon8 over Weapon7

Weapon7Choice.disable()

Weapon8Choice.disable() //disables the pressed button, and the button for the opposite weapon

String Weapon7orWeapon8string = "Weapon 8" //The string gets the value of the chosen weapon, so it can be updated to database

WeaponchoiceCount = WeaponchoiceCount + 1 //the counter increases each time when the choice between two weapons is made

end if

if WeaponchoiceCount = 8 then //as there are 8 pairs of 16 weapons, therefore when the variable equals to 8, all the choices are made

QuitButton.enable() //enables the quit button

end if

if QuitButton = action.pressed then

STATEMENT DatabaseUpdate //opens the statement, which allows to update the database

query UPDATE Database //the query to update the database

WeaponChoiceCondition = TRUE //sets the condition equal to TRUE, which allows to access the WeaponPurchase menu in BuyMenu

Weapon1or2Database = Weapon1orWeapon2string //saves the value of the string in the database, so only one of two weapons will

//be available in the WeaponPurchase menu

Weapon3or4Database = Weapon3orWeapon4string

...

Weapon7or8Database = Weapon7orWeapon8string

end query

end STATEMENT

open BuyMenu() //opens the Buy Menu

close WeaponChoice() //closes the WeaponChoice Menu

end if

end class

# Weapon purchase

class WeaponPurchase() //class of a menu where you can buy a weapon

STATEMENT DatabaseSelectstmt //opens a statement, which allows to work with the database

SELECT DatabaseImport //selects needed data from database

resultSet Side //selects the side that the player have chosen for the game

resultSet Weapon1or2Database //selects the weapon choices from the previous class

resultSet Weapon3or4Database

...

resultSet Weapon7or8Database

resultSet Weapon1Inventory //selects the weapon that the user currently has in his inventory

resultSet Weapon2Inventory

resultSet Grenade1Inventory //selects the grenades that the user currently has in his inventory

...

resultSet Grenade4Inventory

resultSet ArmorInventory //selects the type of armor that the user currently has in his inventory

resultSet fullArmor //selects whether the user has the full armor (not damaged) in his inventory or not

resultSet zeusInventory //selects whether the user has zeus in his inventory or not

resultSet kitsInventory //selects whether the user has kits in his inventory or not

resultSet Balance //selects the current balance of the player

resultSet RoundsWonT //selects the number of rounds that was won by the T side

resultSet RoundsWonCT //selects the number of rounds that was won by the CT side

resultSet RoundsCount //selects the number of rounds that were played overall

resultSet MolotovBuy //checks whether the grenade was bought in this round or not (when coming back to buy menu)

resultSet HEGrenadeBuy

...

resultSet SmokeBuy

end SELECT

end STATEMENT

Text Field TFWeapon1 = set text Weapon1Inventory //text fields display the items in the inventory

Text Field TFWeapon2 = set text Weapon2Inventory

Text Field TFZeus = set text zeusInventory

Text Field TFGrenade1 = set text Grenade1Inventory

Text Field TFGrenade2 = set text Grenade2Inventory

...

Text Field TFGrenade4 = set text Grenade4Inventory

Text Field TFKits = set text kitsInventory

Text Field TFArmor = set text ArmorInventory

Text Field TFRoundCount = set text Weapon1Inventory //text fields display the information about the game

Text Field TFBalance = set text Balance

Text Field TFRoundsWonT = set text RoundsWonT

Text Field TFRoundsWonCT = set text RoundsWonCT

label lSide = set text Side //Displays the side that the player is playing for

button Weapon1NameBuy //buttons to buy a weapon

button Weapon2NameBuy

button Weapon3NameBuy

...

String Weapon1NameString //the name of the purchased weapon to be stored in mySQL

String Weapon2NameString

...

button QuitWeaponPurchaseMenu //allows to quit the weapon purchase menu

if Grenade1Inventory.equals(not null) && ... Grenade4Inventory.equals(not null) then //statement used to disable the buttons used to purchase any grenades, as 4 is the maximum number of the grenades in the inventory

button Molotov.disable()

...

button HEGrenade.disable()

end if

if Side = TSide then //enables only the buttons for weapons available for the T side

enable buttons weapons for T Side

disable buttons weapons for CT Side

enable buttons weapons.(weaponChoiceMenu) //enables only the buttons for chosen weapons

end if

if Side = CTSide then //enables only the buttons for weapons available for the CT side

enable buttons weapons for CT Side

disable buttons weapons for T Side

enable buttons weapons.(weaponChoiceMenu) //enables only the buttons for chosen weapons

end if

if balance < weapon1price then //if the player does not have enough balance to purchase a specific weapon, disable the button

button Weapon1NameBuy.disable()

end if

...

if MolotovBuy = "True" then //disables the button of the grenade that was purchased before

button Molotov.disable()

end if

...

if Weapon1NameBuy = action.pressed then //the weapon was purchased using the button

Balance = Balance - weapon1price //the balance updates

TFBalance = set text Balance //the text field updates

Weapon1Inventory = Weapon1NameString //the database variable updates with the value of the string of the purchased weapon

Weapon1NameBuy.disable() //disables the button, as it was already pressed

end if

if QuitWeaponPurchaseMenu = action.pressed then

STATEMENT DatabaseUpdate //opens the statement, which allows to update the database

query UPDATE Database //the query to update the database

Weapon1Inventory //updates the weapon variable in the database

...

Balance //updates the balance in the database

end query

end STATEMENT

open BuyMenu() //opens the Buy Menu

close WeaponPurchase() //closes the WeaponChoice Menu

end if

end class