# Evaluation

The program-assistant for the Counter-Strike: Global Offensive satisfies the given criteria.

In order to get this assessment, my client and I went through the success criteria:

* *Choose a side - T or CT*

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application, email

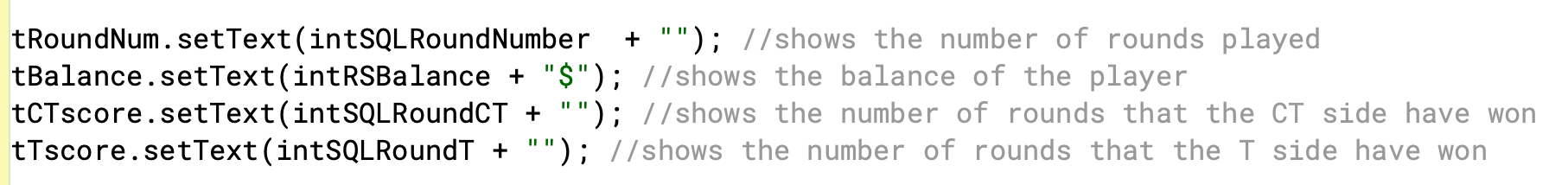
Description automatically generated

The program allows you to do this via a separate menu.

* *Each menu has an information about:* 
  + *The number of the round*
  + *How many rounds has CT and T side won*
  + *The balance of the player*

Table

Description automatically generated



This information is given to the player using text fields and is updated via the MySQL database.

* *Main menu with three buttons: Buy, Game, Result. Each button represent the stages of the round: Buy - what do you want to buy; Game - how is the game going (any kills, was the bomb planted); Result - did you win the round, etc. They have specific order: you cannot access Game and Result, if you haven’t pressed Buy, etc.*

Website

Description automatically generated with low confidence

A picture containing text

Description automatically generated

Graphical user interface, text, application

Description automatically generated

The buttons were made using third-party libraries and GUIs. Each button leads to a separate menu. The condition that one stage cannot be accessed before the other is written in the program class and saved in MySQL.

* *Every menu has a GUI* 
  + *Buy Menu:*
    - *Weapon choice:* 
      * *Some weapons in CS:GO cannot be used in the same game, so you need to choose between them, for example, Desert Eagle or R8 Revolver. Does not apply for all the weapons. After everything is chosen, the button will be disabled till the end of the game. Only works in the first round.*
    - *Weapon purchase menu:* 
      * *Menu where you buy guns and equipment*
    - *Weapon drop/pick* 
      * *Menu used if you wish to drop a gun or to pick a gun from your teammate.*
  + *Game menu:* 
    - *Weapon drop/pick*
      * *Menu used if you wish to drop a gun or to pick a gun.*
    - *Game (buttons)*
      * *Did you make any kills? Input integer*
      * *Which weapon did you use?*
      * *Did you make a teamkill*
      * *Did you plant a bomb, etc.*
      * *Several questions about the result of the game.*
  + *Result menu:*
    - *Displays the results of the round*
* *The program is running in a loop until:*
  + Round count = 30 or
  + *CT won 16 rounds*
  + *T won 16 rounds*

*Text

Description automatically generated with medium confidence*

* All of these criteions have been successfully met using
  + Buttons
  + Labels
  + Text fields
  + Condition variables (for choices)
  + Link to the database
* *All the menus are with buttons; no input from user required.*

This criterion was not fulfilled perfectly, as I decided that in one case input was still a more convenient way than pressing a button.

* *In every menu there is a balance text field, which is the purpose of the program.*
* *In the end of every round the balance is displayed.*
* *Program is connected to mySQL and all the menus (classes) take variables from there.*

All of these criteria are met.

# End user suggestions:

I sent my user the program and he used it for a week and sent his feedback. You can see our dialogue in the appendix section.

My user has encountered the problem that he has to start from the very first round when starting the program. The main events of the game, when both teams have enough balance and weapons for equal play, start around round 4, but in order to get there you have to start at the very beginning and skip to that point.

I see a way of solving this problem. For example, you could make a button at the start of the program that asks whether the user wants to start from the very first round, or from a specific round of the game. If the user selects the option from a particular round, a menu should appear with input where the user can enter the balance score, the rounds won, the number of rounds played and the weapon selected.

Also, the user did not quite like the design. This problem is solved simply by adding a background picture and increasing the size of the fields in the class.

# Personal areas of improvement:

First of all, design. Although my program is user-friendly, the design of my program could have been better and more game-like. For example, add weapon icons, a background like some kind of map, and make each menu look the way it does in the game.

I also think my program would be much better if the user didn't have to click anything at all, just watch the program run. This might be possible if I had enough programming skills to synchronize my program with the real game so that the user could calculate their in-game budget in real time.

Word count: 375