

RELATÓRIO DE SDAC

TRABALHO FINAL DE SDAC



21\04\2022

TEAM WORK

CHOOSE CORRECTLY

ORLANDO SANTOS Nº7

JOSÉ COUTO Nº6

PEDRO TEODORO Nº9

11ºJ

**1-Anteprojeto:**

1,1-Em que consiste o Grand Theft Code:

Com este trabalho em grupo realizado em âmbito com a disciplina de Sistemas digitais e Arquitetura de Computadores, nós pretendemos realizar um programa em linguagem C, em que consiste em diversas decisões do utilizador.

O utilizador terá neste programa possibilidade de realizar várias escolhas onde cada um dos seus atos terão consequências sendo elas:

Em sua rotina, na sua vida financeira, no seu trabalho e perante na sociedade.

O utilizador terá a possibilidade de personalizar o seu nome no jogo, idade e altura e interagir com diversas escolhas ao longo do jogo.

1.2-Planificação do Grand Theft Code:

Em termos de estrutura do programa, nós pretendemos utilizar a estruturas de escolhas ou decisão, repetição ou ciclos, passagem de parâmetros, strings e arrays.

Estas estruturas serão importantes para o desenvolvimento das decisões e das consequências que o utilizador levará ao longo da sua jogatina.

1.3-Tipos de decisões ao longo do programa:

No início do jogo o utilizador/player irá começar com 10.000€ na conta para iniciar a sua vida com a sua personagem fictícia.

Após personalizar seu nome, seu sobrenome, sua idade, sua altura, o utilizador poderá escolher uma profissão fixa onde irá ter um ordenado conforme a sua escolha.

O utilizador deparar-se com escolhas como sair a noite, ir à praia, comprar um carro desportivo ou ir para o trabalho estas escolhas irão ter consequências financeiras.

Alem destes desafios o utilizador terá diversos outros cenários de escolha, como por exemplo deparar-se com a polícia, ser preso, envolver-se numa briga.

O jogo terminará quando de acordo com as escolhas do utilizador, se o utilizador não souber administrar as escolhas corretamente, poderá acabar o jogo com um final falido sem dinheiro ou morto por exemplo.

O JOGO ENCONTRA-SE NA ALPHA, A BETA ESTA QUASE PRONTA.

ÍNDICE:

[1-Anteprojeto: 1](#_Toc5321)

[1,1-Em que consiste o Grand Theft Code: 1](#_Toc31283)

[1.2-Planificação do Grand Theft Code: 1](#_Toc24003)

[2-Introdução: 3](#_Toc1499)

[3-Estrutura do Código: 4](#_Toc10926)

[3.1-PERSOLANIZAÇÃO DA PERSONAGEM: 5](#_Toc25744)

[3.2-ESCOLHA DA PROFISSÃO: 6](#_Toc32455)

[3.3-DEPARAÇÃO DO JOGADOR COM A POLÍCIA: 7](#_Toc21947)

[4-Indicação de elementos do código e Dificuldades: 9](#_Toc10376)

[5-Conclusão 12](#_Toc4050)

**2-Introdução:**

Neste relatório iremos abordar um código que tem a ver com um roleplay de uma espécie de “Grand Theft Auto”, mas em modo de código.

Iremos apresentar algumas funcionalidades, e algumas dificuldades ao longo do código com os erros que foram introduzidos e as respetivas resoluções e também implementações que não foram implementadas com razões para tal esses erros.

Uma imagem com texto

Descrição gerada automaticamente

**3-Estrutura do Código:**

Como já aprendemos o Código C, portanto colocamos esse tipo de código para o nosso projeto final que esta completa.

Como pode ver esta imagem numa parte do código é feita pelo código C que é obviamente diferente em todos os códigos. (fig.1)(fig1.1)

Uma imagem com texto

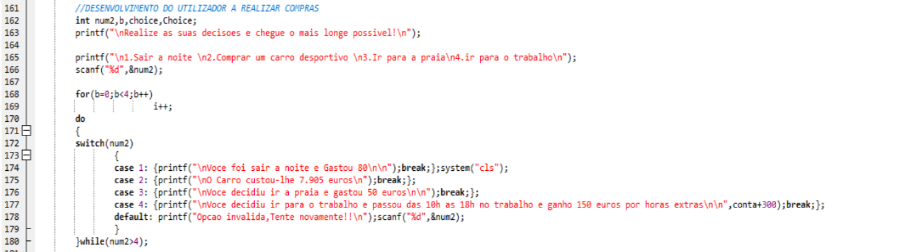
Descrição gerada automaticamente

Uma imagem com texto

Descrição gerada automaticamenteO Jogador poderá consultar o manual do jogo e até mesmo acabar o jogo logo no início do jogo, se pretender sair.

**3.1-PERSOLANIZAÇÃO DA PERSONAGEM:**

A seguinte imagem já mostra o menu do código para dar indicações e regras para cumprir neste roleplay (fig.2)



Uma imagem com texto

Descrição gerada automaticamenteSe o jogador escolher iniciar o Jogo digitando 1, como Primeiro passo terá de criar um personagem a tua escolha, ou seja, a tua identidade pessoal, com o teu nome e sobrenome, e idade (Ex:18+) e a tua altura (Ex:1.75+). (Fig.2.1)

Uma imagem com texto

Descrição gerada automaticamente

**3.2-ESCOLHA DA PROFISSÃO:**

Após customizar a personagem o jogador Escolhera a profissão da personagem, seja um lixeiro, um programador, um mecânico, um funcionário de McDonalds ou até mesmo o desemprego, mas se quiseres escolher mais tarde, é com o jogador. (Mas se escolher um número acima de 6 é o número invalido.) (Fig.3)(Fig 3.1).

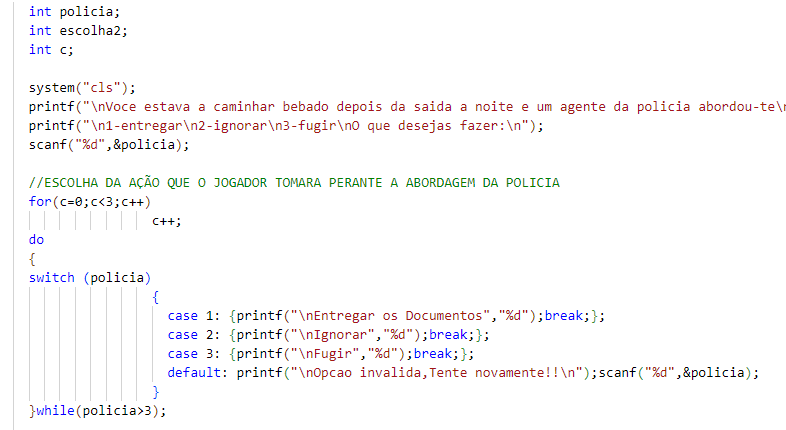
Uma imagem com texto

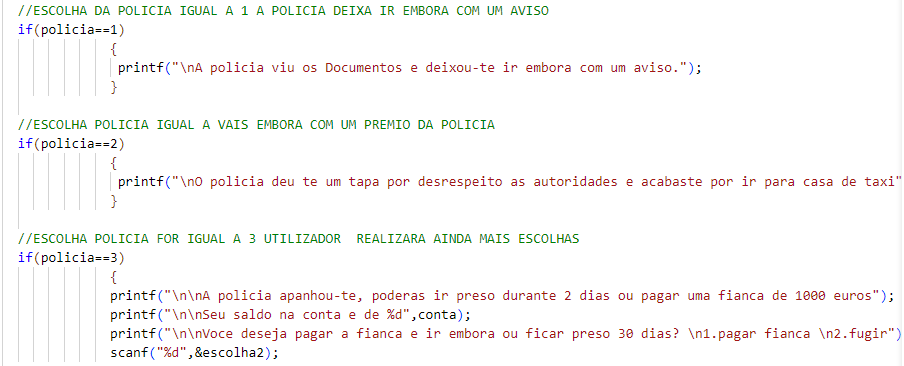
Descrição gerada automaticamenteUma imagem com texto

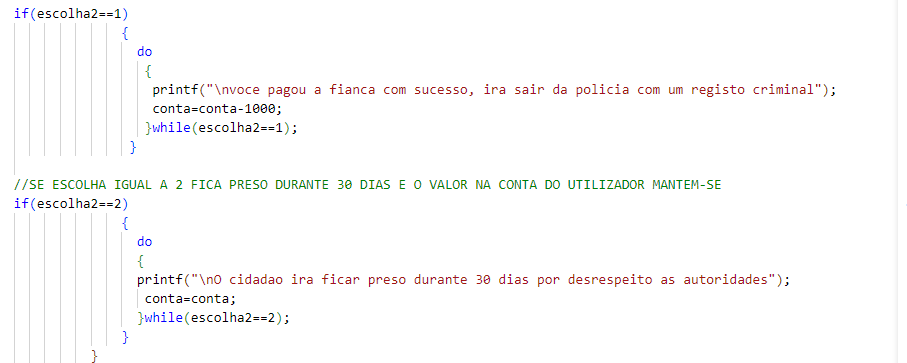
Descrição gerada automaticamente

**3.3-DEPARAÇÃO DO JOGADOR COM A POLÍCIA:**

Existe uma parte no jogo em que o utilizador irá deparar-se com a polícia e dependendo de sua ação, o utilizador irá deparar-se com ir preso, seguir a vida calmamente, ou levar uma advertência, podendo até chegar ao Game Over.(Fig 4)(Fig 4.1)(Fig 4.2).

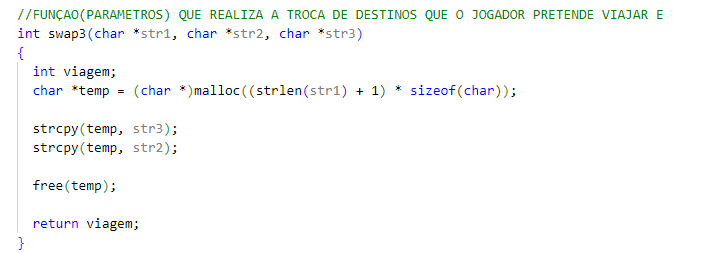


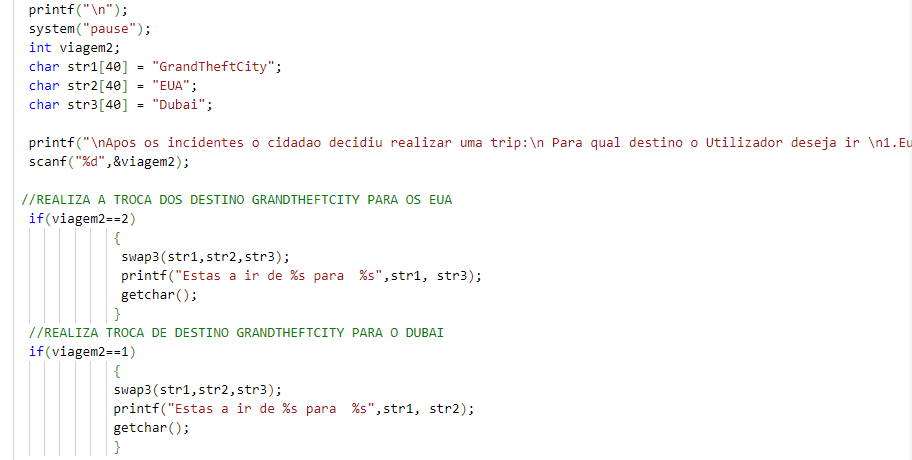


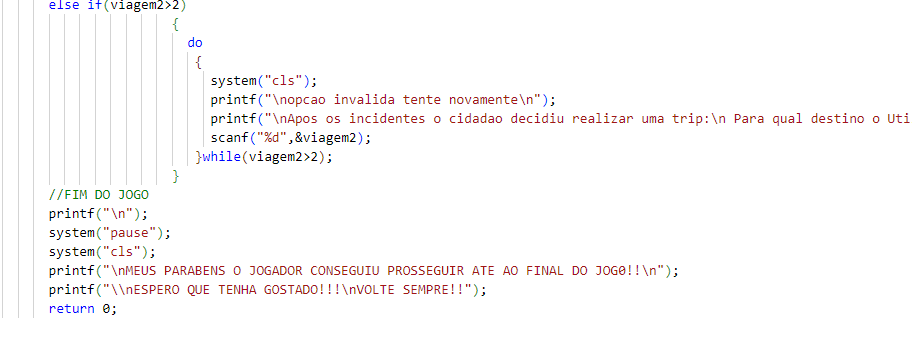


**3.4-REALIZAÇÃO DE UMA VIAGEM A ESCOLHA DO UTILIZADOR**

Após todos os eventos o jogador poderá por escolha decidir viajar para algum lugar, sendo eles os EUA ou o DUBAI , deixando GRANDTHEFTCITY.(Fig 5)(Fig 5.1).







Levando o utilizador ao final do jogo após estes 3 os obstáculos presentes(Fig5.2).

**4-Indicação de elementos do código e Dificuldades:**

Neste código nos recorremos a utilização de Estruturas de decisão, repetição ou ciclos e a passagem de parâmetros.

Uma imagem com texto

Descrição gerada automaticamenteUma imagem com texto

Descrição gerada automaticamenteUm dos Pontos mais fotes que nõs acabamos por refrorçar foram os switches seguidoes de if’s encadeados, para reforçar a decisão e o acontecimento da decisão.

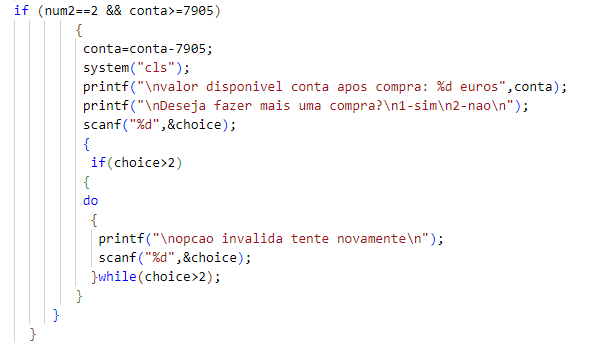
Como podem ver nestas duas imagens a utilizaçãom das estruturas de decisão.

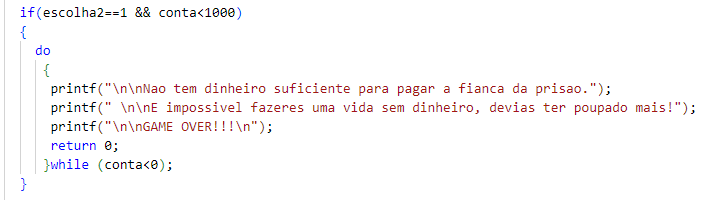
Apesar desta realização e deste padrão, acabamos por perceber que ele não servia para as demais ações pretendidas no código, levando a diversas falhas e tentativas, desde a utilização de apenas ium do while para todas as decisões(Não correndo nada como o esperado e resultando na perda do código).

A utilização de varios If’s encadeados para os acontecimentos(o que acabou por correr melhor, pois facilitou chegar a decisão sem erros e sem implicações nas outras escolhas presentes).

Estas falhas com os switches apareceram mais na parte das compras , pois houveram situações em que o utilizador chegava a numeros negativos na conta e o jogo prosseguia sem problemas, então para evitar este tipo de situação recorreu se a junção de acontecimentos através do indicador”&”.

Este tipo de Estrutura esta muito presente na situação realizando a proteção de erros.

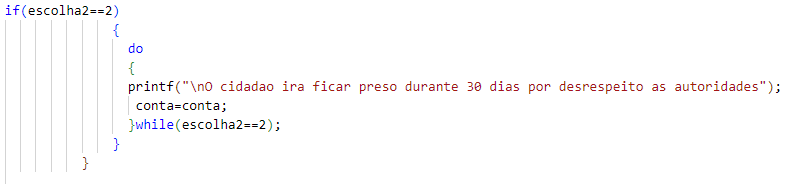


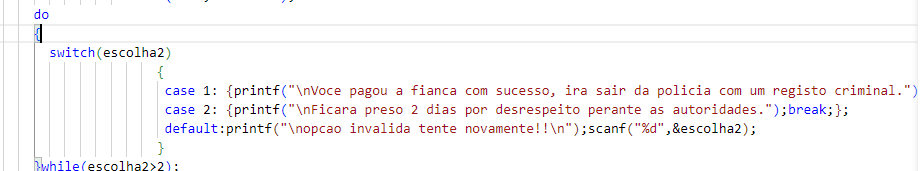


Mais um dos problemas ao longo do desenvolvimento do código foi o facto de ao o utilizador digitar um número superior a seleção, mesmo com os if’s e os do while no código, o utilizador prosseguia no jogo sem nenhum aviso do número digitado ser um número inválido.

Estes tipos de erros são erros lógicos, que mesmo o programa parecendo funcional ele apresenta erros lógicos na estrutura, estando as vezes certas instruções antes declaradas ou depois, tendo de ser localizadas em locais diferenciado.

Para o arranjo desta solução, foi-se utilizada a independência dos If’s, cada um com o seu “do while”, não ficando as mesmas dependentes de apenas uma estrutura de repetição e sim de várias, tal como a adição de estruturas de Switches, reforçando as escolhas e seguido de if’s, reforçando as hipóteses ao fazer a escolha.





# 5-Conclusão

Com este trabalho aprendemos a programar em C e a juntar várias estruturas em apenas um código, apesar das dificuldades e das horas passadas a frente do monitor a concertar os erros, o projeto do trabalho ficou como o pretendido, realizando o prometido e levando ao utilizador a ter uma pequena imersão nesta espécime de mini jogo na janela de comandos.

Foi-se exigida uma prática por parte do(s) aluno(os) no trabalho e um conhecimento extensivo na programação em c, ficando ainda alguns pormenores por executar no código, mas com o principal objetivo cumprido.

Foi se concluído que programar é algo que exige um certo conhecimento acerca de erros lógicos, tal como um raciocínio matemático, também que erros lógicos são as partes que exigem mais raciocínios e que quanto mais organizado e mais limpo um código, mais percetível a localização de seus erros e de argumentos em falta.