

Faculdade Engenharia Universidade do Porto



Gamers' Gym

Projeto Final Laboratório de Computadores

Turma 6 – Grupo 5

Realizado por:

- Filipe Esteves (up202206515@up.pt)
- Miguel Mateus (up202206944@up.pt)
- Álvaro Torres (up202208954@up.pt)

Índice

Introdução.....	3
1. Instruções de Utilização.....	4
1.1 Menu Inicial.....	4
1.2 Jogo <i>Reaction Time</i>.....	5
1.3 Jogo <i>Aim Trainer</i>.....	6
2. <i>Project Status</i>.....	7
2.1 Dispositivos.....	8-9
3. Organização/estrutura de código.....	10-12
4. Conclusões.....	13

Introdução

No nosso jogo, o “Gamers’ Gym” pretendemos implementar dois jogos: um jogo de reação e outro jogo de "aim trainer". Ambos os jogos são interativos e têm como objetivo melhorar as habilidades de entusiastas de videojogos entre outros. Iremos ter um menu onde o jogador pode escolher o jogo. Após a escolha do jogo o utilizador navega para o jogo escolhido podendo iniciar o jogo.

1. Instruções de Utilização

1.1 Menu Inicial

No início do jogo, o menu inicial é mostrado. Através de dois botões o jogador pode escolher entre dois jogos: *Reaction Time* ou *Aim Trainer*. A ação de cada botão pode ser acedida por movimentos do rato, através de um clique com o botão esquerdo sob a área pretendida do botão pretendido.



Fig. 1 – Menu Inicial

1.2 Jogo *Reaction Time*

Neste jogo, o jogador deverá acionar o jogo através do teclado. Uma vez acionado uma tela vermelha com o texto “Wait for green...” aparece. O jogador deverá então esperar que a tela fique verde e aí poderá clicar para medir o seu tempo de reação, que será mostrado na tela. Caso o jogador clique na tela cedo demais, terá de tentar novamente.

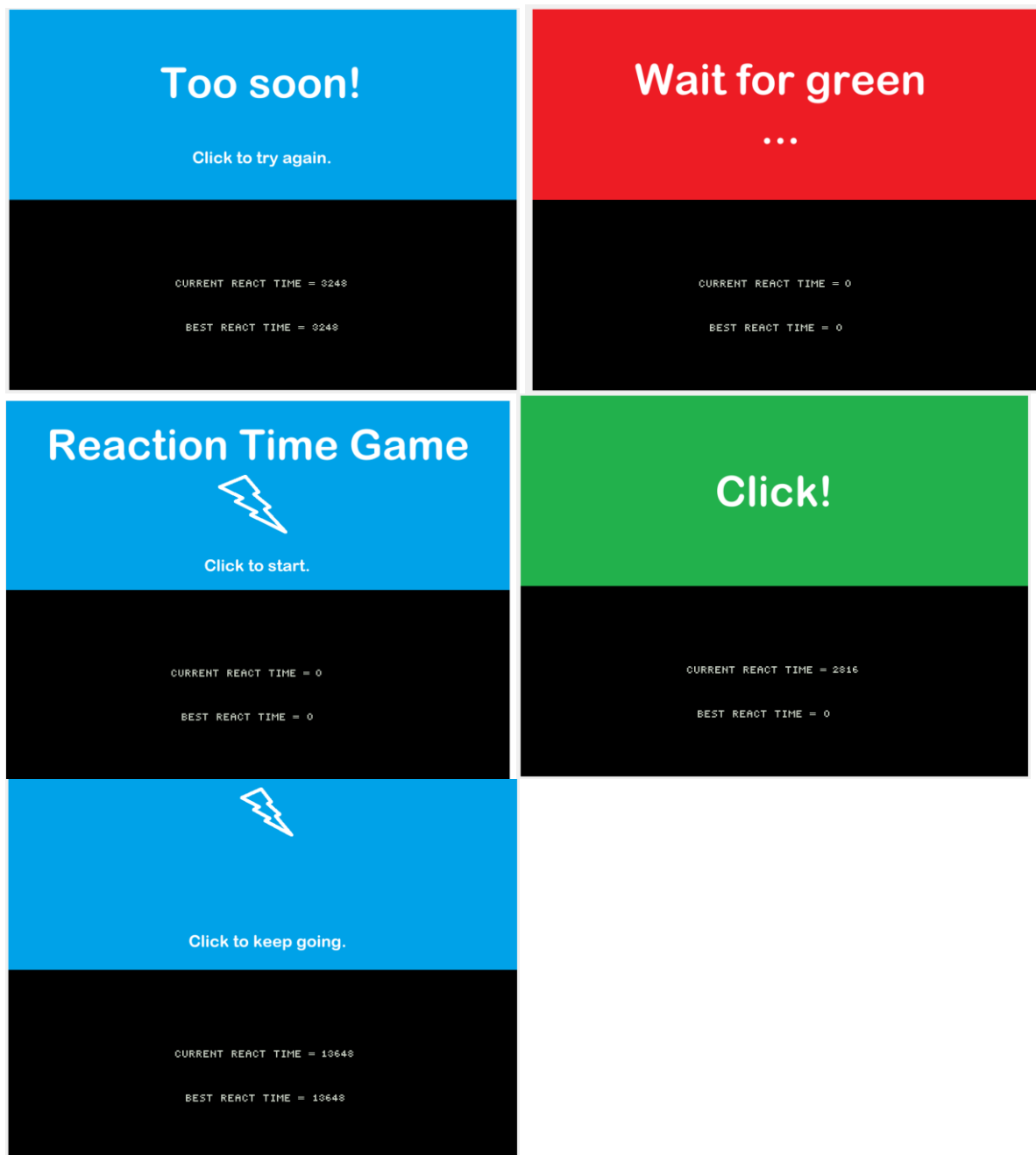


Fig. 2-6 – Jogo *Reaction Time*

1.3 Jogo *Aim Trainer*

Neste jogo, o jogador, através de um menu, poderá escolher entre dois modos de jogo: *Classic* ou *Time Attack*. No jogo clássico, o jogador terá de acertar em todos os alvos, clicando sobre cada alvo no botão esquerdo do rato. O jogador começa o jogo com 3 vidas e vai perdendo vidas caso clique numa área da tela que não seja um alvo. O *score* é o tempo que o jogador demorou a acertar em todos os alvos. Após clicar num alvo outro alvo é instantaneamente gerado numa posição aleatória na tela. No modo *Time Attack*, o jogador tem 1 minuto para tentar acertar no máximo de alvos possíveis, ficando registado no seu *high score*, a quantidade de alvos em que foi capaz de acertar. Após finalizar ou perder o jogo, o utilizador regressa sempre ao menu do jogo *Aim Trainer* automaticamente, podendo voltar a optar por um dos modos de jogo ou então voltar ao menu principal.

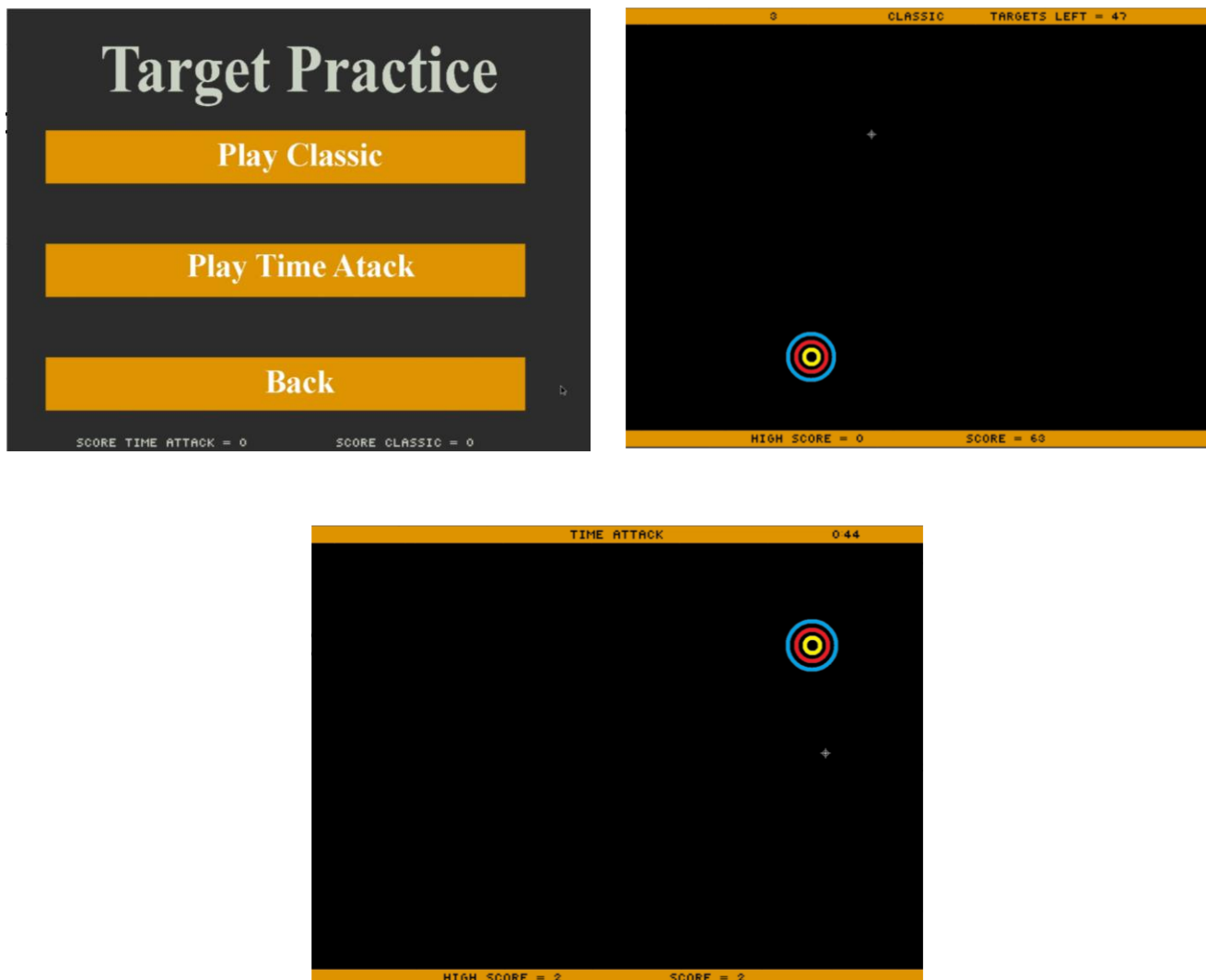


Fig. 7-9 – Jogo *Aim Trainer*

2. *Project Status*

Dispositivo	Funcionalidade	Implementação
Timer	Contador de tempo do jogo <i>Time Attack</i> e <i>Classic</i> do <i>Aim Trainer</i> ; Contador de tempo para o jogo de reação	Sim (interrupts)
KBD	Usado para no jogo <i>Reaction Time</i> acionar/parar o contador do tempo de reação	Sim (interrupts)
Mouse	Usado para navegar entre menus e para selecionar os jogos e modos de jogo; Usado no jogo <i>Aim Trainer</i> para “acertar” nos alvos, utilizando o botão do lado esquerdo	Sim (interrupts)
Video Graphics	Mostrar toda a interface com os jogadores, os menus, os jogos em si	Sim (sem interrupts)
RTC	Usado no menu principal para mostrar a hora e data atual	Sim (sem interrupts)

2.1 Dispositivos

Graphics Card

Todos os gráficos presentes no nosso projeto foram transformados de PNG para XPM com o programa GIMP.

Fundamental para a visualização do jogo e para a mudança de cores e gráficos do nosso jogo tanto no jogo de reação como no "aim trainer" onde um alvo mudará de local na tela. É fundamental para mostrar toda a interface, todos os menus e o jogo em si. Usamos o VBE mode 0x14C, com resolução de ecrã de 1152x864 com direct color (32 ((8:)8:8:8)), e como tal permite 2^{32} cores diferentes. Foi iniciada uma implementação de buffering que não chegou a ser finalizada no produto final.

A implementação da configuração e funcionalidades referentes ao video graphics encontram-se no ficheiro graphics.c.

Keyboard

O teclado é usado para no jogo *Reaction Time* acionar/parar o contador do tempo de reação (space bar).

A implementação das configurações, funcionalidades e interrupções do teclado estão nos ficheiros keyboard.c e KBC.c.

Mouse

No nosso projeto o mouse é usado para navegar entre menus e para seleccionar os jogos e modos de jogo, usando o botão do lado esquerdo do mouse. Este também usado no jogo *Aim Trainer* para "acertar" nos alvos, utilizando o botão do lado esquerdo, que quando clicado sobre um alvo o mesmo desaparece e outro é gerado numa posição aleatória da tela.

A implementação das configurações, funcionalidades e interrupções do mouse estão no ficheiro mouse.c.

RTC

O Real-Time Clock é usado para mostrar a hora e data atual no menu principal. A implementação das configurações, funcionalidades do RTC estão nos ficheiros rtc.c e gamesprites.c (método draw_main_menu).

Timer

O timer é fundamental para o funcionamento do nosso jogo. O timer é a base para o jogo *Reaction Time*. Depois de o jogador acionar o jogo, após um tempo definido a tela muda de cor (de vermelho para verde) e o jogador tem de pressionar a barra de espaço para contabilizar o tempo da sua reação. O timer será fundamental para contabilizar o tempo de começo do jogo tanto como a medição do tempo de reação. Já no jogo *Aim Trainer* o timer é usado para contabilizar o tempo restante de jogo no modo de jogo *Time Attack* e para contabilizar o tempo que o jogador demora a acertar em todos os alvos (score) no modo de jogo *Classic*.

A implementação das configurações, funcionalidades e interrupções do timer estão no ficheiro `timer.c`.

3. Organização/estrutura de código

Timer – 10%

Este módulo contém as funções desenvolvidas no Lab2 das aulas práticas de LCOM que foram reutilizadas e integradas no nosso projeto. Estas funções gerem tudo relacionado com o timer. Este módulo encontra-se em `src/Controlers/Timers`.

KBD Module – 5%

Este módulo contém as funções desenvolvidas no Lab3 das aulas práticas de LCOM que foram reutilizadas e integradas no nosso projeto. Estas funções gerem tudo relacionado com o keyboard. Este módulo encontra-se em `proj/src/Controlers/Keyboard`.

Mouse – 10%

Este módulo contém as funções desenvolvidas no Lab4 das aulas práticas de LCOM que foram reutilizadas e integradas no nosso projeto. Estas funções gerem tudo relacionado com o mouse. Este módulo encontra-se em `proj/src/Controlers/Mouse`.

Video Graphics – 10%

Este módulo contém as funções desenvolvidas no Lab5 das aulas práticas de LCOM que foram reutilizadas e integradas no nosso projeto. Estas funções gerem tudo relacionado com a video graphics. Este módulo encontra-se em `proj/src/Controlers/Graphic`.

RTC – 2%

Este módulo contém as funções desenvolvidas no âmbito da integração do RTC no nosso projeto. O RTC é usado como um relógio para que o utilizador tenha informação sobre a hora e data atuais. Este módulo encontra-se em

proj/src/Controlers/RTC.

Utils Module – 1%

Este módulo contém as funções desenvolvidas no Lab2 das aulas práticas sobre o timer que posteriormente continuaram a ser usadas nos Labs seguintes. Este módulo encontra-se em proj/src/Controlers.

Menu Module – 7%

Este módulo contém o código referente aos menus do jogo. Através de cliques do mouse nos vários botões o jogador pode escolher jogos e modos de jogo, tal como navegar entre telas. Este módulo encontra-se em proj/src/GameControlers: gamesprites.c.

Sprites Module – 4%

Este modulo tem todas as funções responsáveis por criar e desenhar os sprites a partir dos XPM's gerados de PNG's. Este módulo encontra-se em proj/src/GameControlers: sprites.c and gamesprites.c.

Reaction Time Module – 22%

Este módulo contém o controlador da lógica do jogo, REACTION TIME, incluindo as funções fundamentais para seu funcionamento. Ele inclui um randomizador de tempo de espera e a interação com a estrutura do usuário quanto ou high score.

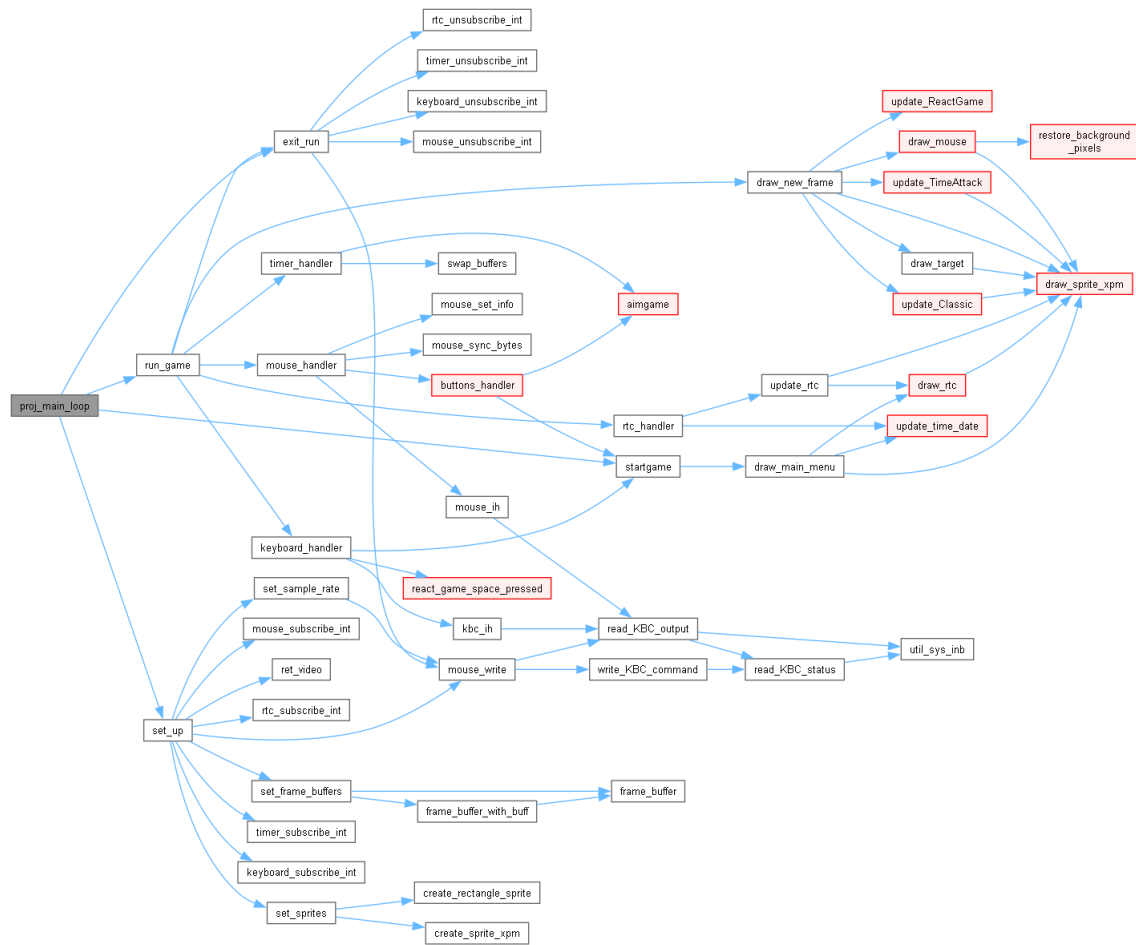
Aim Trainer Module – 22%

Este módulo contém o controlador da lógica do jogo, TIME ATTACK e CLASSIC, incluindo as funções fundamentais para seu funcionamento. Ele inclui um randomizador de posição para o alvo e a interação com a estrutura do usuário, tanto nos high scores quanto na quantidade de vidas.

Main – 10%

Este módulo representa a entrada principal do nosso programa, contendo o loop principal do projeto. Este módulo encontra-se em proj/src: main.c

Function call graph



4. Conclusões

Em suma, retrospectivamente estamos todos bastante satisfeitos e orgulhosos do nosso trabalho. Conseguimos implementar tudo aquilo a que nos propusemos a implementar neste projeto e concretizamos a nossa ideia inicial. Para além de desenvolvermos as nossas *skills* e conhecimento como programadores, desenvolvemos também competências de trabalho de equipa e de elaboração de relatórios.