

Curso 8016 - Licenciatura Tecnologias da Informação

Ano letivo 2017/2018

1º Semestre

## **PROJETO TEMÁTICO EM PLANEAMENTO DE PROJETOS INFORMÁTICOS**

### **Bomber Trap**

#### **Autores:**

Ana Castro

Diogo Simões

Ricardo Balreira

Ricardo Alves

Tiago Silva

**Grupo** III

ESTGA – Universidade de Aveiro

Curso 8016 - Licenciatura Tecnologias da Informação

Ano letivo 2017/2018

1º Semestre

## **PROJETO TEMÁTICO EM PLANEAMENTO DE PROJETOS INFORMÁTICOS**

### **Bomber Trap**

#### **Autores:**

Ana Castro

Diogo Simões

Ricardo Balreira

Ricardo Alves

Tiago Silva

**Grupo** III

#### **Orientadora:**

Zulmira Pereira

ESTGA – Universidade de Aveiro

## Resumo

Este projeto visa explorar mecanismos e formas de desenvolver um algoritmo com o intuito de criar um jogo lúdico, respeitando o objetivo esperado. Com recurso a algumas ferramentas, tais como: sustentabilidade do planeamento de projetos, design visual e o software Scratch para a criação do jogo, é possível, assim, reunir o conjunto de características ideais para o desenvolver de um jogo com uma boa estrutura, quer em termos gráficos quer em termos funcionais.

O trabalho tem como principal objetivo recriar uma das clássicas sagas da série “Bomberman” que consiste, de uma forma geral, em eliminar o adversário traçando um plano estratégico utilizando a “bomba” como único recurso. De forma a tornar o jogo mais competitivo, o complemento adicionado ao jogo parte do pressuposto de obter uma chave secreta situada num lugar aleatório do campo do jogo e, deste modo, percorrer o trajeto necessário até à porta que indica, precisamente, o vencedor da partida. O jogo também permite ao(s) jogador(es) poder desfrutar da partida com um ambiente sonoro e com uma boa qualidade gráfica.

De uma forma objetiva, o trabalho baseia-se na descrição das etapas desenvolvidas, no que respeita ao planeamento do projeto, planificação das tarefas, desenvolvimento do jogo e melhoria do aspeto visual. Consiste também na abordagem aos softwares utilizados no decurso do desenvolvimento do jogo, tomando como explícito a definição do software, fundamentação do motivo pela qual se escolheu o mesmo para determinada intenção no objetivo e vantagens técnicas. Predominam também as dificuldades surgidas no desenvolvimento do algoritmo e a superação desses constrangimentos com soluções advindas durante a elaboração do jogo. Não tomando como menos relevante, o domínio e a distribuição das tarefas, os requisitos funcionais e não funcionais e a descrição das etapas realizadas no algoritmo do jogo são também referidos na estrutura do relatório, não desvalorizando qualquer aspeto que tenha sido crucial na elaboração do trabalho.

# Índice

Índice de figuras.....	ii
Índice de tabelas .....	iii
1. Introdução .....	1
2. Planificação do trabalho .....	2
2.1 Mapa de Gantt previsto .....	2
2.2 Mapa de Gantt executado.....	3
2.3 Levantamento de Requisitos .....	5
2.3.1 Requisitos funcionais e não funcionais .....	5
3. Descrição das atividades desenvolvidas .....	6
3.1 Bomber Trap.....	6
3.1.1 “CodeUA” .....	7
3.1.2 Repositório.....	13
3.1.3 Sincronização com o “CodeUA” .....	14
3.1.4 Microsoft Project .....	16
3.1.5 Scratch.....	20
3.1.6 Photoshop.....	29
4. Análise dos resultados .....	34
5. Conclusão .....	35
6. Bibliografia.....	38

## Índice de figuras

Figura 1 - Mapa de Gantt previsto .....	3
Figura 2 - Mapa de Gantt executado .....	4
Figura 3 - Página inicial do "CodeUA" .....	7
Figura 4 - Criação do novo projeto .....	8
Figura 5 - Visão geral .....	8
Figura 6 - Atividade .....	9
Figura 7 - Tarefas (representação do corpo do projeto) .....	10
Figura 8 - Gráfico de Gantt .....	10
Figura 9 - Calendarização das tarefas .....	11
Figura 10 - Repositório (Git) .....	12
Figura 11 - Configurações .....	12
Figura 12 - Modo de armazenamento da informação de outros sistemas (CVS, Subversion, Bazaar, etc.) .....	13
Figura 13 - Modo funcional do Git .....	14
Figura 14 - Criação da pasta .....	15
Figura 15 - Colocação dos ficheiros .....	16
Figura 16 - Endereço do "CodeUA" .....	16
Figura 17 - Modelos incorporados do MS Project .....	17
Figura 18 - Planeamento de projetos no MS Project .....	18
Figura 19 - Linha cronológica .....	18
Figura 20 - Atribuição de recursos (Folha de Recursos) .....	19
Figura 21 - Algoritmo nível "Difícil" .....	22
Figura 22 - Vista Tela do jogo .....	23
Figura 23 - Algoritmo do Menu Inicial .....	24
Figura 24 - Vista Menu Inicial .....	25
Figura 25 - Algoritmo Movimento do <i>Bot</i> .....	26
Figura 26 - Vista Jogo com Bot .....	27
Figura 27 - Vista de tela e algoritmo da "chave" e "porta" .....	28
Figura 28 - Criação do "Bomberman" azul .....	29
Figura 29 - Criação do "Bomberman" vermelho .....	30
Figura 30 - Criação dos <i>Bot's</i> .....	30
Figura 31 - Bloco explosivo .....	31
Figura 32 - Criação da bomba .....	31
Figura 33 - Criação da explosão .....	32
Figura 34 - Criação da porta .....	32
Figura 35 - Criação da chave .....	33
Figura 36 - Versão final do jogo .....	33

## **Índice de tabelas**

Tabela 1 - Requisitos funcionais.....	6
Tabela 2 - Requisitos não funcionais.....	6

## **1. Introdução**

No âmbito do Projeto Temático em Planeamento de Projetos Informáticos, foi proposto desenvolver um videojogo, com recurso a uma plataforma ou software de criação de jogos. Os professores responsáveis pela orientação dos grupos foram os professores João Paulo Hespanha e Zulmira Pereira. A plataforma era ao critério de cada grupo formado por cinco elementos, sendo exigido o mais completo possível na implementação da demo. Era recomendado utilizar a plataforma "Scratch", pois é baseada numa metodologia de programação mais simples. O objetivo estava em torno da sustentabilidade do projeto, isto é, numa boa fundamentação do jogo e num bom planeamento das tarefas, utilizando algumas plataformas disponibilizadas pela escola tais como o "CodeUA" e o "MS Project". Optou-se por desenvolver uma versão inspirada na série de videojogos designada por "Bomberman" que consiste, basicamente, em eliminar o(s) oponente(s) com um único recurso – a bomba.

Neste relatório serão apresentados os procedimentos realizados ao longo do planeamento e execução do jogo começando com a planificação do trabalho (tarefas desenvolvidas ao longo do projeto, esquematização das mesmas e das atividades do projeto em cronogramas, tipos de requisitos associados ao desenvolvimento do jogo e folha de recursos), seguida de uma descrição pormenorizada das atividades desenvolvidas (tipos de softwares usados, fases/etapas durante o mesmo e onde está presente também uma descrição do jogo em si) que nos permitirá analisar os resultados e, por fim, refletir criticamente e concluir sobre o projeto.

## **2. Planificação do trabalho**

A fase de planeamento deve contemplar várias etapas onde se definem os objetivos gerais e de segunda ordem, o tipo de tarefas, os recursos necessários à concretização dos objetivos e das tarefas, e a pesquisa. É na fase de execução/implementação que se aplicam todas as tarefas e requisitos estabelecidos na fase de planeamento. Nesta fase desenvolve-se, neste caso, o jogo usando o software Scratch. Nela também se efetuam as atualizações das tarefas recorrendo a ajustes consoante o decorrer do projeto em si. Por fim, na fase de finalização recorre-se a uma verificação geral de todo o projeto, validando-o e procedendo á realização do relatório final.

Neste trabalho, na fase de planeamento, mais concretamente durante o planeamento de tarefas e elaboração de cronogramas, foi necessário recorrer ao *software* Microsoft Project - software de gestão de projetos produzido pela Microsoft.

### **2.1 Mapa de Gantt previsto**

No âmbito da fase de planeamento, a elaboração de um cronograma é uma forma de organizar e gerir as tarefas durante a execução de um projeto. Consiste na determinação da melhor forma de posicionar as tarefas ao longo do tempo de acordo com a duração das mesmas, das relações de precedência entre elas e dos prazos a cumprir. Com base nisto, foi elaborado um cronograma correspondente a este projeto no qual estão presentes as atividades previstas bem como o tempo previsto de execução de cada uma delas (figura 1).



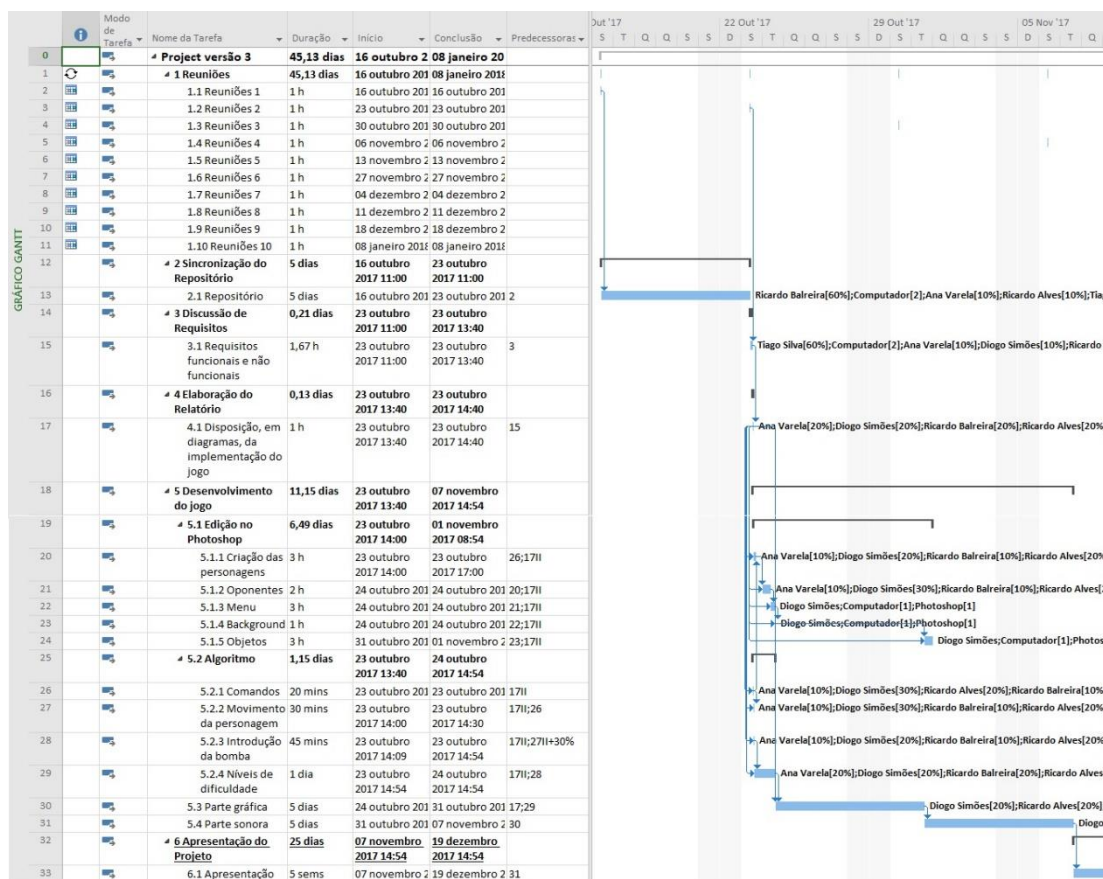
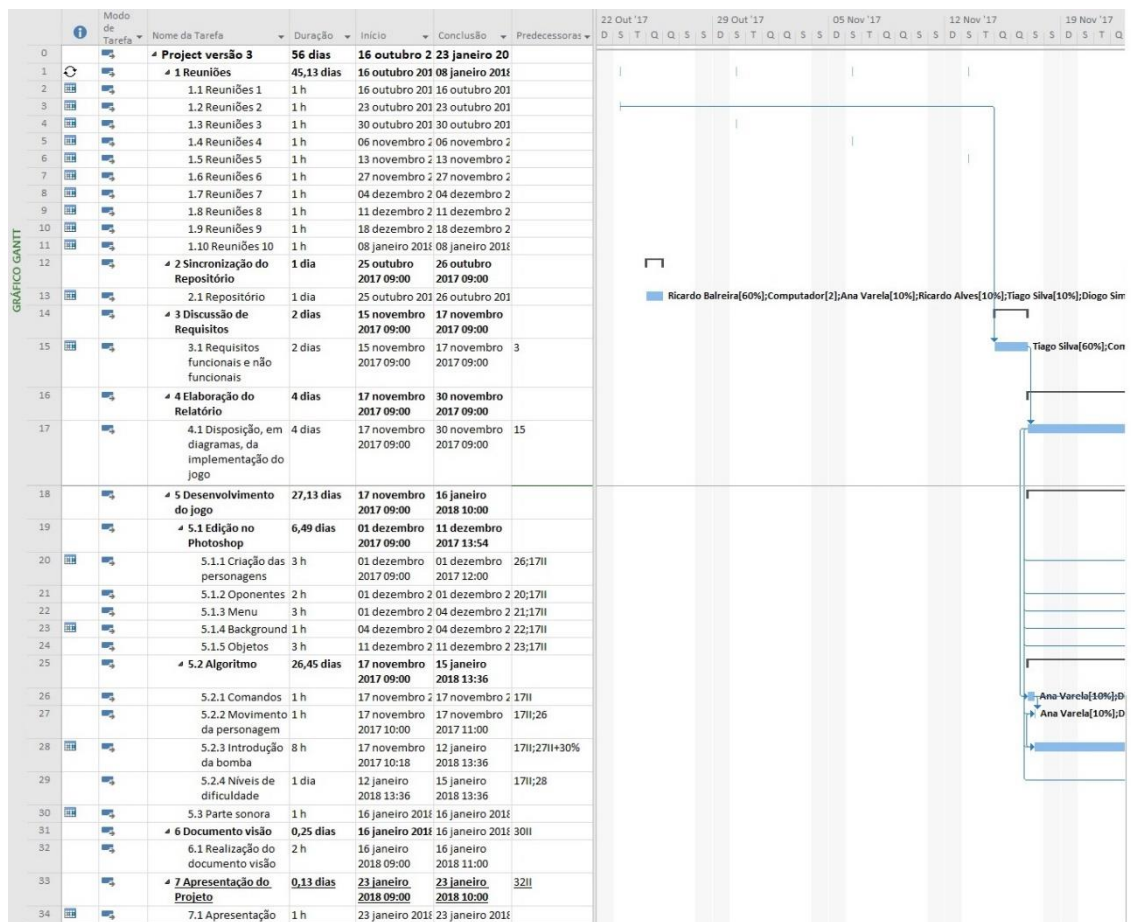


Figura 1 - Mapa de Gantt previsto

## 2.2 Mapa de Gantt executado

Esta fase consistia em aplicar todas as tarefas estabelecidas nas fases iniciais do planeamento. À medida que se iam realizando e implementando as tarefas, observou-se que eram necessários ajustes relativamente às mesmas. No cronograma da figura 2 estão as tarefas atualizadas de acordo com imprevistos que iam decorrendo ao longo do projeto.



**Figura 2 - Mapa de Gantt executado**

## **2.3 Levantamento de Requisitos**

Um requisito é uma condição necessária e indispensável colocada sobre um serviço ou sistema. O levantamento de requisitos está associado ao processo de descobrir quais são as operações que o sistema deve realizar e quais são as restrições que existem sobre essas mesmas operações.

### **2.3.1 Requisitos funcionais e não funcionais**

Os requisitos tanto podem ser funcionais e não funcionais. Os primeiros são definidos como as funcionalidades ou atividades que um sistema/*software* deve realizar, já os não funcionais devem conter elementos específicos, tais como: descrição da tarefa a ser executada pelo *software*, origem do requisito e o seu utilizador, relação da passagem de informação entre o *software* e o utilizador e, se existirem, algumas restrições lógicas associadas à tarefa. Dentro dos requisitos não funcionais estão incluídos os requisitos de desempenho, de interface, operacionais, de recurso, de verificação e de aceitação, entre outros.

Os requisitos não funcionais estão relacionados com os requisitos funcionais e indicam como o sistema/*software* deve ser feito e como deve funcionar, ou seja, são os critérios que qualificam os requisitos funcionais.

Em seguida estão apresentadas tabelas que contêm os requisitos funcionais e não funcionais alusivos ao desenvolvimento do jogo. (tabela 1 e 2 respetivamente). Todos os requisitos não funcionais mencionados na tabela 2 estão relacionados com os requisitos funcionais explícitos na tabela 1.

**Tabela 1 - Requisitos funcionais**

Id	Requisitos funcionais
RF1	Menu inicial do jogo (escolher nº de jogadores, dificuldade, iniciar)
RF2	Os bonecos devem ser capazes de se mover para todos os lados
RF3	Existem <i>bots</i> que se movem de maneira aleatória pelo mapa
RF4	Existem partes do mapa que podem ser destruídas e outras não
RF5	Ao clicar em certa tecla o boneco tem de largar uma bomba
RF6	A bomba tem que explodir
RF7	A explosão não pode destruir partes do mapa
RF8	A explosão destrói <i>bots</i> e o boneco
RF9	O jogo acaba quando o boneco for destruído ou se destruir todos os <i>bots</i>
RF10	Existir vários níveis de dificuldade em que se aumenta a velocidade dos <i>bots</i> , da explosão e o número de <i>bots</i>
RF11	Destruindo todos os <i>bots</i> passa-se de nível

**Tabela 2 - Requisitos não funcionais**

Id	Requisitos não funcionais
RNF1	Menu de configurações (tirar ou não o som da musica e os efeitos sonoros)
RNF2	Menu de pausa
RNF3	Quando sair do menu de pausa dar 3 segundos antes de começar
RNF4	Jogo pode correr em <i>full screen</i>

### 3. Descrição das atividades desenvolvidas

#### 3.1 Bomber Trap

Na execução do projeto, estiveram disponíveis como suporte e sustentabilidade do planeamento e execução do jogo quatro plataformas – “CodeUA”, repositório (“Git”), “MSProject”, “Scratch”. Antes de passar à descrição das etapas realizadas no decurso do projeto, é importante conhecer as mesmas referidas.

### 3.1.1 "CodeUA"

Esta plataforma, uma ferramenta disponibilizada pelos serviços académicos, permite criar um projeto novo ou utilizar um já existente tendo acesso a alguns parâmetros relacionados com o planeamento e gestão de projetos. Permite ao gestor ou ao grupo estabelecer um planeamento conciso e organizado do projeto, utilizando os recursos disponibilizados pelo site.

O conjunto de imagens apresentadas seguidamente definem de uma forma mais objetiva o propósito da plataforma.

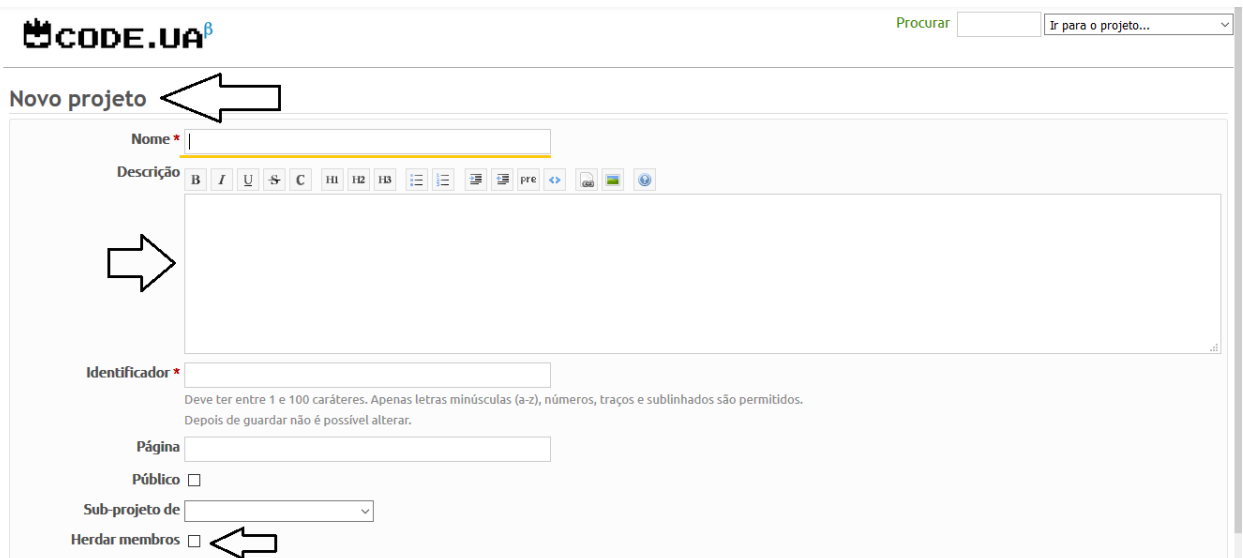
Em primeiro lugar, pela figura 3 é possível observar a página inicial (*home page*) da plataforma onde refere de forma simples o seu objetivo (sublinhado a cor de laranja). Para ter acesso aos projetos e subprojetos basta selecionar a opção indicada.



**Figura 3 - Página inicial do "CodeUA"**

Em segundo lugar, para ser possível criar um novo projeto é preciso atribuir um conjunto de parâmetros importantes para que o mesmo seja reconhecível. Tal como mostra a figura 4, os campos obrigatórios são o

nome e um identificador (sem segundas alterações). Os restantes campos são ao critério do utilizador (facultativos).



**Novo projeto**

Nome \*

Descrição

Identificador \*

Página

Público ☐

Sub-projeto de

Herdar membros ☐

**Figura 4 - Criação do novo projeto**

O "CodeUA" apresenta 8 secções ou categorias. A primeira de todas, a Visão geral, permite controlar as tarefas que estão pendentes e concluídas divididas pelo seu tipo (figura 5). No lado direito é possível ver os membros que pertencem ao projeto definidos hierarquicamente.



**Visão geral**

Tarefas

	abertos	fechados	Total
Bug	1	1	2
Funcionalidade	14	1	15
Suporte	2	0	2

Membros

Manager: Ana Rita Varela Castro, Diogo Alexandre das Neves Simões, Ricardo André dos Santos Balreira, Ricardo Filipe Almeida Alves, Tiago Marques da Silva

Developer: Ana Rita Varela Castro, Ricardo André dos Santos Balreira, Ricardo Filipe Almeida Alves, Tiago Marques da Silva

Professor: Zulmira Maria dos Santos Pereira

**Figura 5 - Visão geral**

A categoria Atividade permite também ao gestor ou ao grupo ter um maior controlo sobre todas as alterações feitas dentro do projeto. Organizadas por ordem cronológica, sendo um campo importante no mundo do planeamento (figura 6).



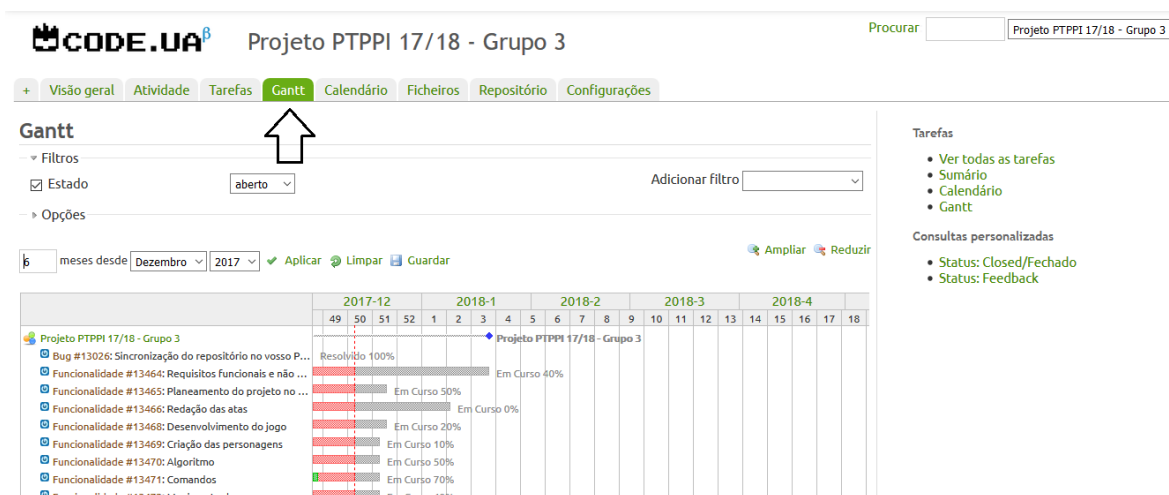
**Figura 6 – Atividade**

Sendo um dos campos com mais relevância na plataforma, a categoria Tarefas permite ao grupo adicionar tarefas que pretendam realizar consoante o projeto. Basta selecionar Nova tarefa, tal como mostra a figura 7, e cumprir os campos exigidos (nome da tarefa, descrição, atribuição da tarefa, estado, prioridade, % completo, ficheiros, observadores).



**Figura 7 - Tarefas (representação do corpo do projeto)**

A plataforma mostra também um gráfico de “Gantt” que permite modelar e ilustrar as tarefas necessárias para realizar um projeto representando-as num diagrama está também incluído na plataforma. Sempre que se insere uma nova tarefa, o gráfico atribui-a ao esquema estando sincronizada com o campo Tarefas (figura 8).



**Figura 8 - Gráfico de Gantt**

Tal como acontece no gráfico de Gantt, as tarefas estão associadas ao calendário o que permite ver o dia a que foram atribuídas (figura 9).






**Figura 9 - Calendarização das tarefas**

Para a inserção de tarefas, é exigida uma ferramenta que permita adicionar e remover ficheiros no "CodeUA". Através da ligação indicada na figura 10, é possível sincronizar a plataforma com o computador pessoal, tal como veremos mais à frente.

**CODE.UA<sup>β</sup>** Projeto PTPPI 17/18 - Grupo 3 Procurar  Projeto PTPPI 17/18 - Grupo 3 ▾

+ Visão geral Atividade Tarefas Gantt Calendário Ficheiros **Repositório** Configurações

Git  

**ptppi1718\_\_git @ master** Estatísticas | Ramo: master ▾ | Revisão:


Nome	Tamanho
Apresentação Scratch.pptx	2,31 MB
Ata 1ª semana.pdf	684 KB
Ata 2ª semana.pdf	674 KB
Ata 3ª semana.pdf	553 KB
Ata 4ª semana.pdf	527 KB
Ata 5ª semana.pdf	522 KB
Ata 6ª semana.pdf	657 KB
Ficheiros adicionados através do Git	
Documentação do Desenvolvimento versão 1.docx	12,6 KB
Documentação do Desenvolvimento.docx	1,76 MB
Estrutura Relatório.docx	13,2 KB
Jogo (Scratch) versão 1.sb2	93,1 KB
Objetivo do jogo.docx	56,8 KB
Planeamento Grafismo e Sonoro.nntx	74,1 KB

**Figura 10 - Repositório (Git)**

Por último, no campo das Configurações tem-se acesso às informações do projeto, membros, como adicionar um novo membro caso o utilizador o pretenda fazer e também como adicionar um novo repositório ou uma nova categoria (Figura 11).

**CODE.UA<sup>β</sup>** Projeto PTPPI 17/18 - Grupo 3 Pro

+ Visão geral Atividade Tarefas Gantt Calendário Ficheiros Repositório **Configurações**

**Configurações** 

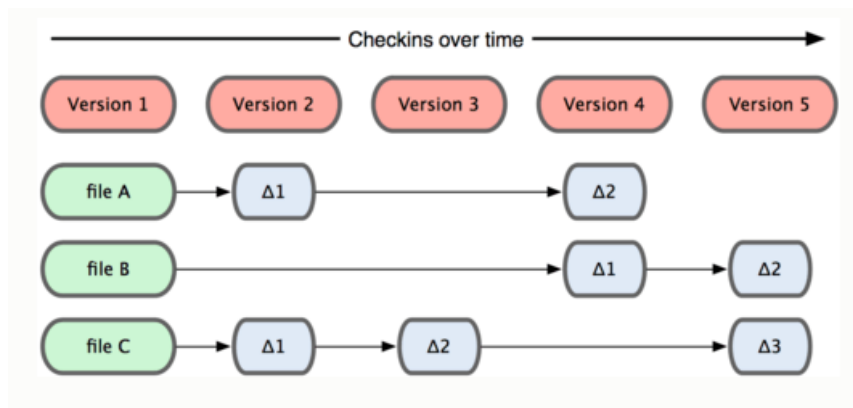
Novo membro

Utilizador / Grupo	Funções
Diogo Alexandre das Neves Simões	Manager
Ana Rita Varela Castro	Manager, Developer
Ricardo André dos Santos Balreira	Manager, Developer
Ricardo Filipe Almeida Alves	Manager, Developer
Tiago Marques da Silva	Manager, Developer
Zulmira Maria dos Santos Pereira	Professor

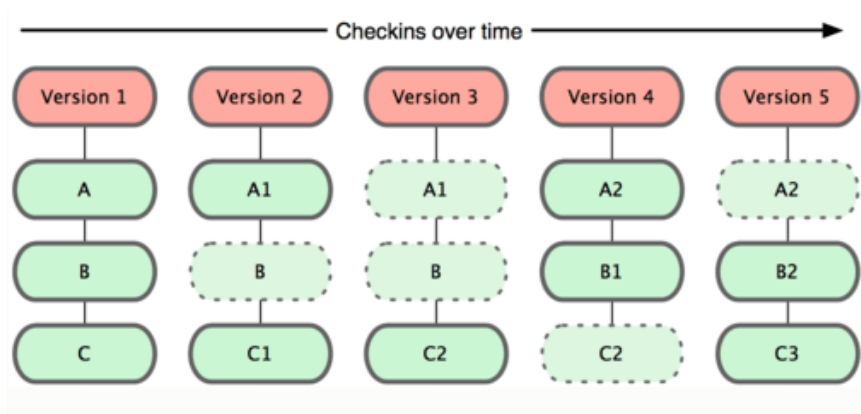
**Figura 11 - Configurações**

### 3.1.2 Repositório

Uma das principais funções da plataforma descrita anteriormente é ser capaz de armazenar ficheiros externos num lugar, sendo uma fonte de segundo recurso caso haja perda de dados por parte do utilizador – repositório. É também de fácil acesso, representando assim a estrutura principal do projeto. Para que tal aconteça, ou seja, para ter acesso ao repositório é necessário sincronizar o computador pessoal com a plataforma. Com isto, utilizar-se-á a ferramenta “Git” que consiste, basicamente, num sistema de controlo e de arquivos que armazena os ficheiros numa referência atribuída pelo utilizador. Uma vez que grande parte deste tipo de ferramentas trata os arquivos como uma lista de mudanças (Figura 10), o Git funciona de um modo diferente: cada vez que se armazena um novo ficheiro este tira a informação dos ficheiros anteriores, como uma foto instantânea, e guarda-a no seu armazenamento (Figura 11).



**Figura 12 - Modo de armazenamento da informação de outros sistemas (CVS, Subversion, Bazaar, etc.)**



**Figura 13 - Modo funcional do Git**

### 3.1.3 Sincronização com o "CodeUA"

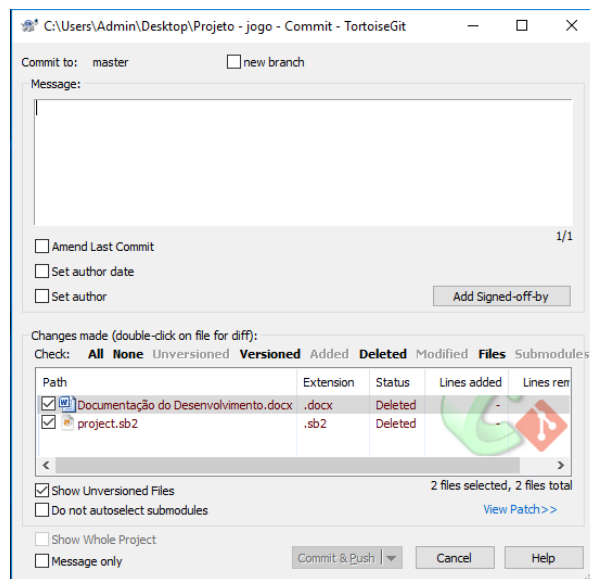
As imagens seguintes servem como uma breve instrução da forma como se cria um repositório associado ao computador pessoal.

Primeiramente irá ser preciso criar uma pasta num local do computador, guardar um ficheiro na mesma e, a partir do botão do lado direito do rato com a pasta seleccionada, seleccionar a opção "Git Create repository here" (figura 14).

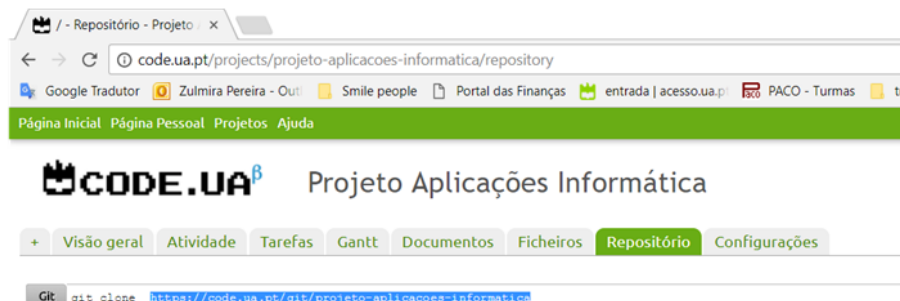


**Figura 14 - Criação da pasta**

O passo seguinte envolve apenas, com o ficheiro na pasta que servirá de repositório, enviá-lo para o endereço associado à plataforma. Desta forma, atribui-se uma mensagem identificando o ficheiro, adiciona-se a data e autor (facultativo) e a opção Commit & Push que permitirá guardar as alterações no repositório local e no local remoto (figura 15). Por último, associa-se o endereço do "CodeUA" ao destino (Destination) do repositório, selecciona-se "OK" e está concluído o envio (figura 16).



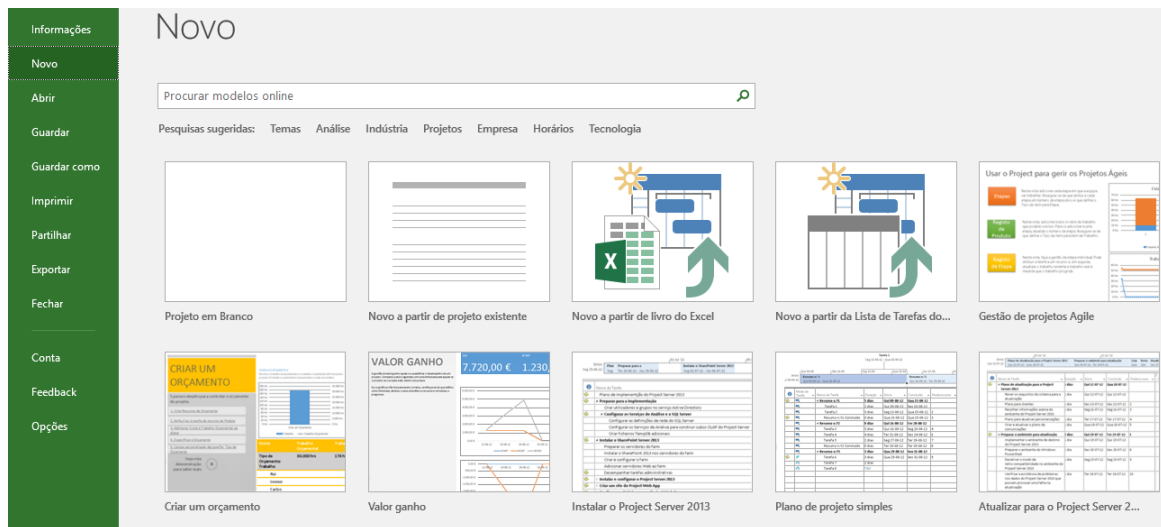
**Figura 15 - Colocação dos ficheiros**



**Figura 16 - Endereço do "CodeUA"**

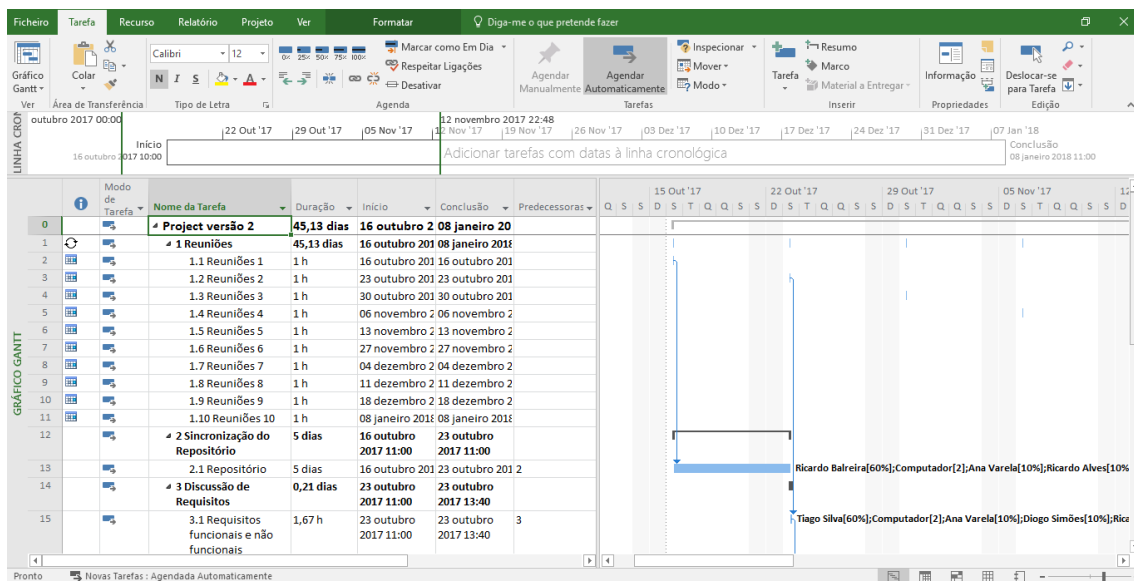
### 3.1.4 Microsoft Project

Os modelos incorporados e personalizados (figura 17) utilizam as práticas recomendadas da indústria para ajudar a começar no caminho certo, não sendo necessário criar planos a partir do zero. Neste caso o grupo decidiu começar um projeto em branco e desenvolver a partir desse projeto.



**Figura 17 - Modelos incorporados do MS Project**

Depois de limitar o projeto a nível de prazos, criou-se uma tarefa principal, à qual foi atribuída um calendário com horários de trabalho semelhantes ao de um trabalhador por conta de outrem. Foram-se adicionando mais tarefas, bem como a respetiva duração, agendadas automaticamente, de forma a que as mesmas fossem ajustadas à concretização do objetivo do jogo. Pode-se observar o resultado no cronograma da Figura 18.



**Figura 18 - Planeamento de projetos no MS Project**

As linhas cronológicas (figura 19) permitem conferir rapidamente todas as atividades do projeto, desde tarefas a marcos futuros. Podem ser personalizadas para apresentar dados específicos e partilhar as mesmas facilmente com os intervenientes do projeto.



**Figura 19 - Linha cronológica**



Após a implementação de predecessoras nas tarefas, para que fosse seguida uma sequência de execução das mesmas, foram-lhes atribuídas certos recursos, previamente definidos na Folha de Recursos (Figura 20).

Alocar os recursos às equipas de projeto adequadas é fundamental para uma gestão de recursos eficaz. É possível criar e editar perfis de recurso com detalhes importantes, como o tipo de funcionário, a função principal, habilidades e experiência, exibir, editar e acompanhar alocações e carga de trabalho de recursos, e comparar a capacidade com a demanda do recurso.

		Nome do Recurso	Tipo	Rótulo Material	Iniciais	Grupo	Unidade Máx.	Taxa Normal	Taxa Trab.	Custo/Utiliz.	Imputar Em	Calendário Base	Código	Atribuir Nova Color.
FOLHA DE RECURSOS	1	Ana Varela	Trabalho		A		100%	0,00 €/h	0,00 €/h	0,00 € Rateado		Calendário		
	2	Diogo Simões	Trabalho		D		100%	0,00 €/h	0,00 €/h	0,00 € Rateado		Calendário		
	3	Ricardo Balreira	Trabalho		R		100%	0,00 €/h	0,00 €/h	0,00 € Rateado		Calendário		
	4	Ricardo Alves	Trabalho		R		100%	0,00 €/h	0,00 €/h	0,00 € Rateado		Calendário		
	5	Tiago Silva	Trabalho		T		100%	0,00 €/h	0,00 €/h	0,00 € Rateado		Calendário		
	6	Computador	Material		C			0,00 €		0,00 € Rateado				
	7	Scratch 2.0	Material		S			0,00 €		0,00 € Rateado				
	8	Photoshop	Material		P			0,00 €		0,00 € Rateado				
	9													

**Figura 20 - Atribuição de recursos (Folha de Recursos)**

### **3.1.5 Scratch**

A plataforma Scratch utiliza uma linguagem de programação visual virada para a programação orientada a objetos. Existem objetos (sprites) que atuam num palco e que são programados combinando vários blocos com diferentes funções, como eventos, controladores, sensores, operadores, etc., que permitem realizar histórias interativas, jogos e animações. Existe também uma biblioteca de “atores” e “cenários”, sendo possível começar com algumas bases.

O site centraliza uma comunidade bastante vasta e dinâmica e oferece uma interface Web para divulgar os projetos, estando disponível em mais de 40 línguas, gratuitamente. Graças à simplicidade de sintaxe na programação, a plataforma pode ser utilizada por todos os alunos, todas as pessoas (diferentes idades, características, níveis de ensino).

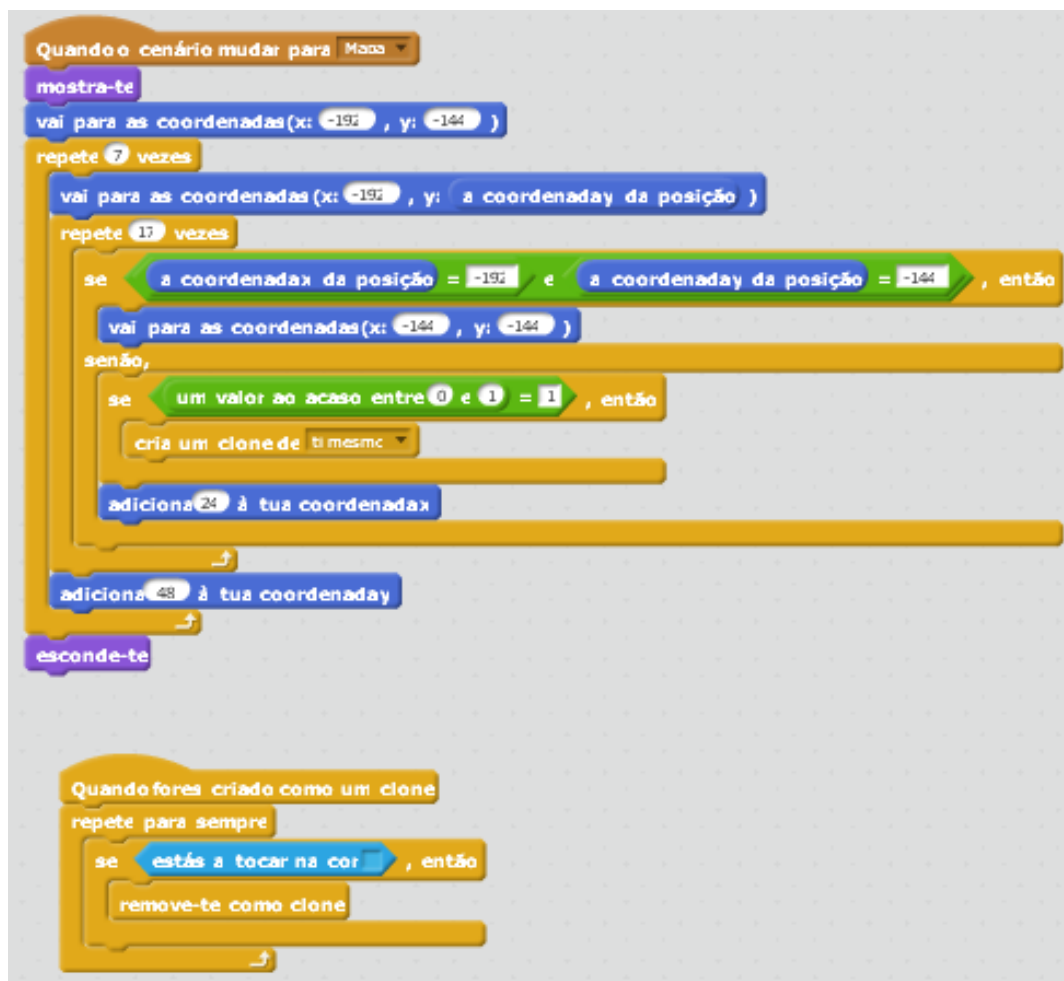
O texto e imagens que se seguem têm como objetivo de explicar pormenorizadamente os procedimentos realizados no decurso do demo.

Começou-se por criar o movimento dos personagens principais. Assim que o utilizador pressiona uma das setas direcionais do teclado, o personagem executa alguns passos na direção estabelecida, guardando na variável “Velocidade Passos (Bomberman) 1”. O ciclo é idêntico para as restantes direções consoante a respetiva indicação do jogador.

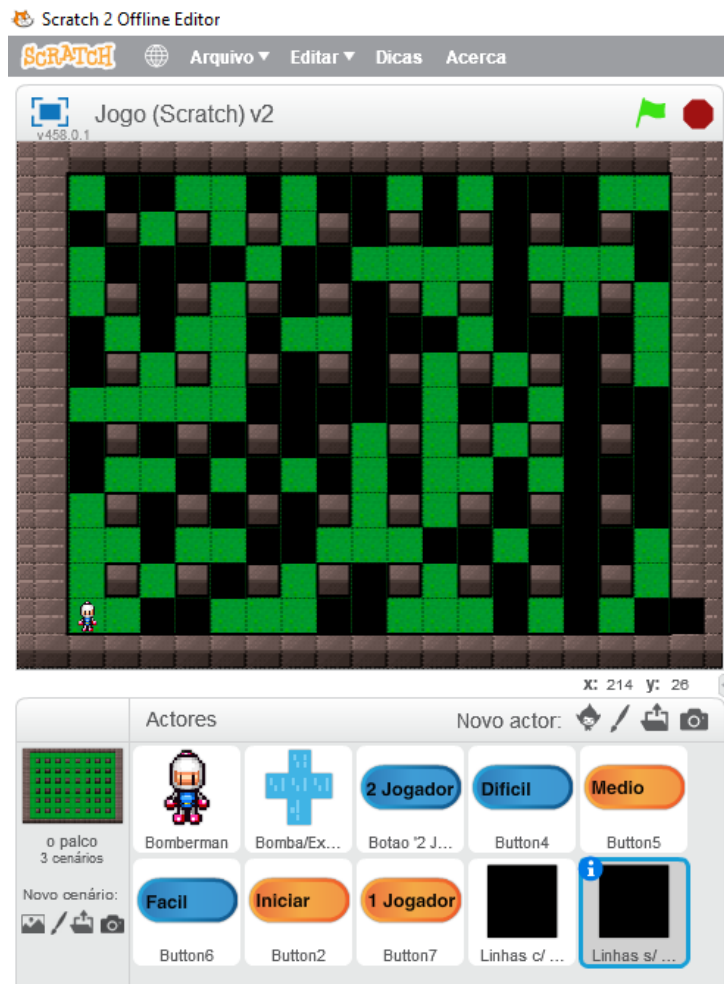
Em seguida, era necessário implementar um novo controlo para a plantação da bomba, o que se optou pela tecla “Espaço”, pelo simples motivo de ser uma tecla universal e fácil de interiorizar durante o jogo. Após plantar a bomba, esta desaparece passado um segundo.

Como o campo do jogo é constituído por coordenadas, quanto ao espaço, e a própria bomba não pode ser detonada em qualquer lugar do mapa, definiu-se que, assim que o utilizador pressiona a tecla "Espaço", o programa guarda a posição atual do jogador. Verificam-se então duas condições: se a coordenada  $x$  é divisível por 24 guarda numa variável designada por "X – Posição Bomba 1"; caso contrário, ajusta a coordenada de forma a condição tornar-se verdadeira, encontrando o menor número que seja divisível por 24 a seguir à coordenada verificada, e guarda-a na mesma variável, procedendo ao mesmo contexto na coordenada  $y$ . Após essa pequena alteração a bomba é plantada. A versão possui também dois tipos de objetos dentro do jogo: "Tijolos Permanentes" e "Tijolos Destrutíveis". Os permanentes nunca alteram de posição, independentemente do modo, e nunca são destruídos pelo jogador, sendo essenciais para delimitar o campo do mapa. Já nos destrutíveis, a situação é distinta: podem ser destruídos pelo utilizador e são gerados de modo aleatório nas coordenadas do mapa. Esse modo aleatório funciona de uma forma peculiar: realiza-se sempre um sorteio entre o intervalo de dois números, e, caso o resultado seja 1, o bloco é colocado na posição e procede para a próxima; caso contrário, passa imediatamente para a seguinte. Entre os três modos ("Fácil", "Médio", "Difícil") o intervalo entre os valores é menor do "Fácil" para o "Difícil" aumentando a probabilidade do resultado ser 1 e colocar o bloco na posição em questão. Os tijolos nunca são gerados em 6 posições específicas que são as posições iniciais dos jogadores.

A seguinte imagem (figura 21) diz respeito ao algoritmo desenvolvido que afirma o que foi descrito anteriormente. Neste caso, para o modo "Difícil", o programa começa a gerar os blocos e a contagem das coordenadas e a validar os seguintes dados: caso as coordenadas não sejam iguais a -192 e -144, muda para a posição seguinte; se não executa o sorteio, e se o resultado for 1 implementa o bloco ou então adiciona 24 à coordenada. Após o primeiro ciclo estar concluído, adiciona 48 à coordenada y e regressa para o início do ciclo. O segundo trecho de código verifica a posição da bomba, pois se esta invadir o espaço de um bloco, este é apagado/removido do mapa. A figura 22 está associada ao algoritmo descrito.



**Figura 21 - Algoritmo nível "Difícil"**

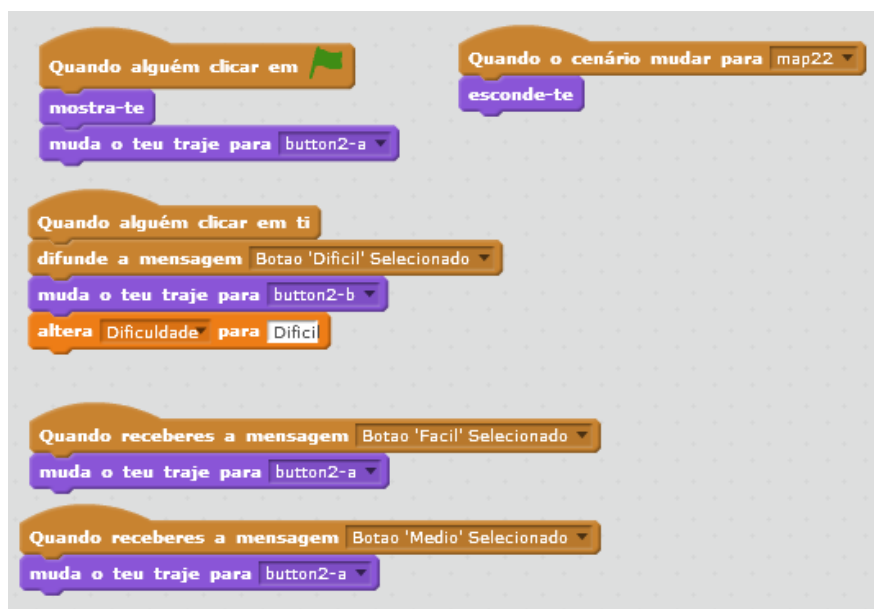


**Figura 22 - Vista Tela do jogo**

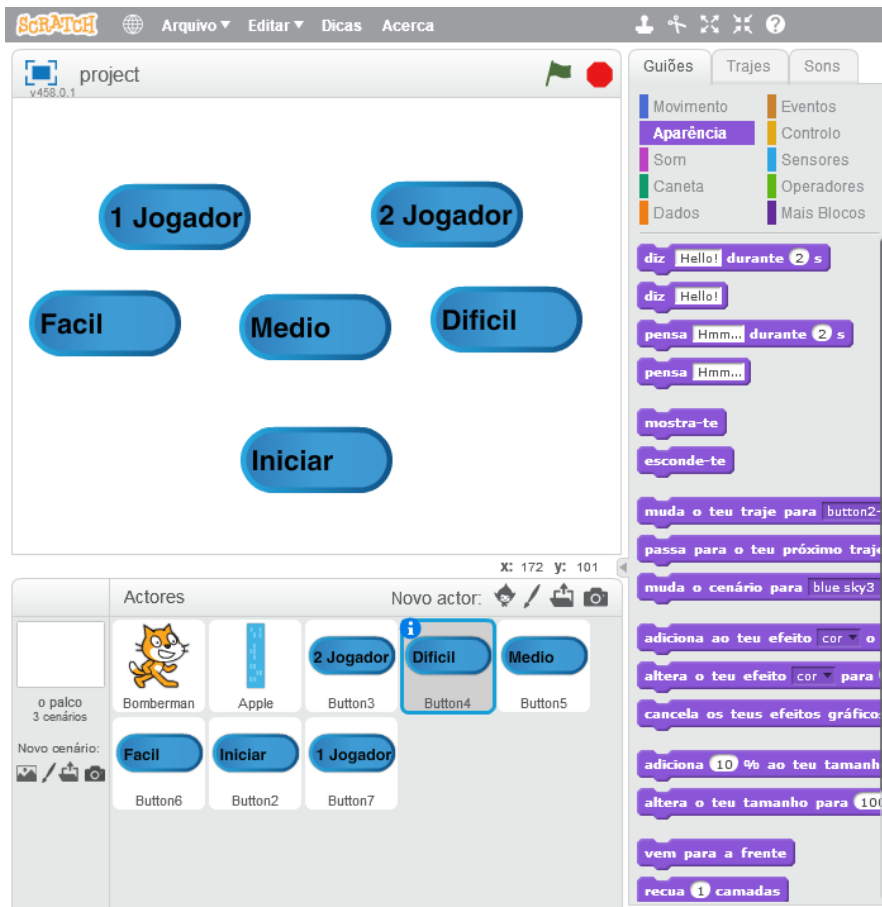
Com o terreno todo gerado podemos agora trabalhar no complemento que pode modificar este ambiente, as explosões. Na posição onde a bomba é plantada é criado um efeito de chama, onde se verificará se podemos criar uma animação de explosão diretamente ao lado dessa. Se o efeito de chama estiver a invadir o espaço dos tijolos permanentes então esse efeito não excede esse espaço, senão é então criado. E repete isto ciclo quatro vezes cujas direções espalham-se com origem no centro da bomba. Se a chama estiver a tocar em algum tijolo destrutível então esse tijolo é destruído. Passado meio segundo de o último efeito ser gerado apaga-se a chama do mapa.

Criou-se um Menu Inicial onde se escolhe quantos jogadores vão jogar e a dificuldade. Também se criaram todas as variáveis dedicadas à dificuldade.

Como é possível ver pela figura 22, cada opção escolhida pelo utilizador dá direito a uma nova saída.

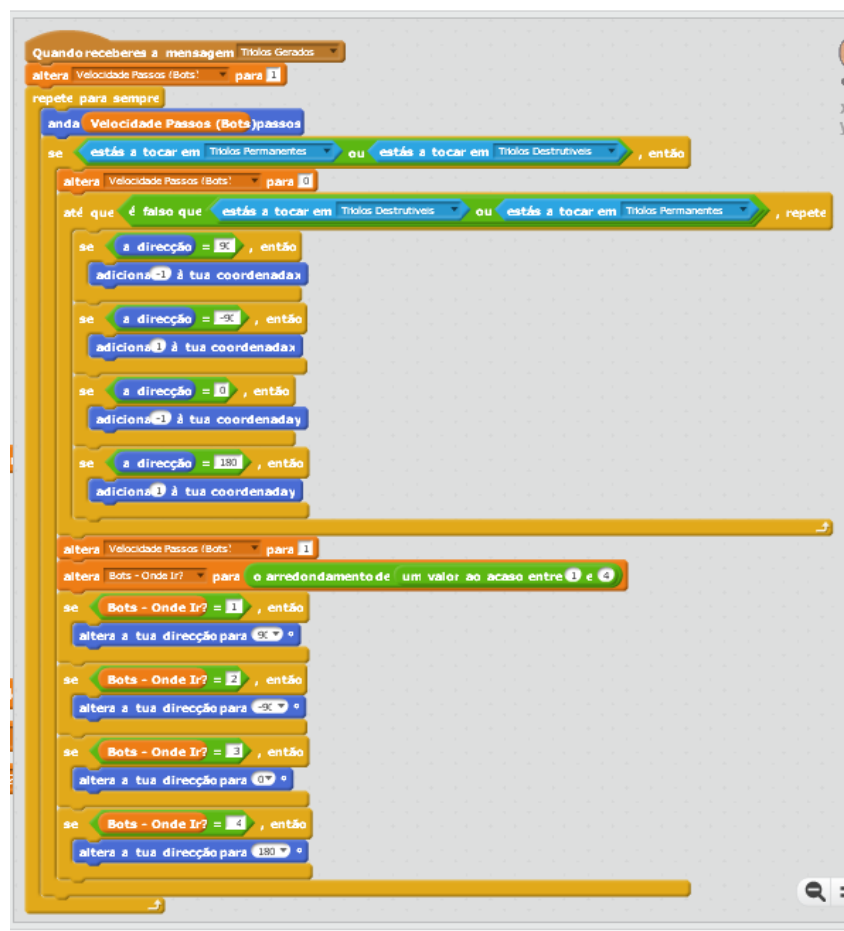


**Figura 23 - Algoritmo do Menu Inicial**



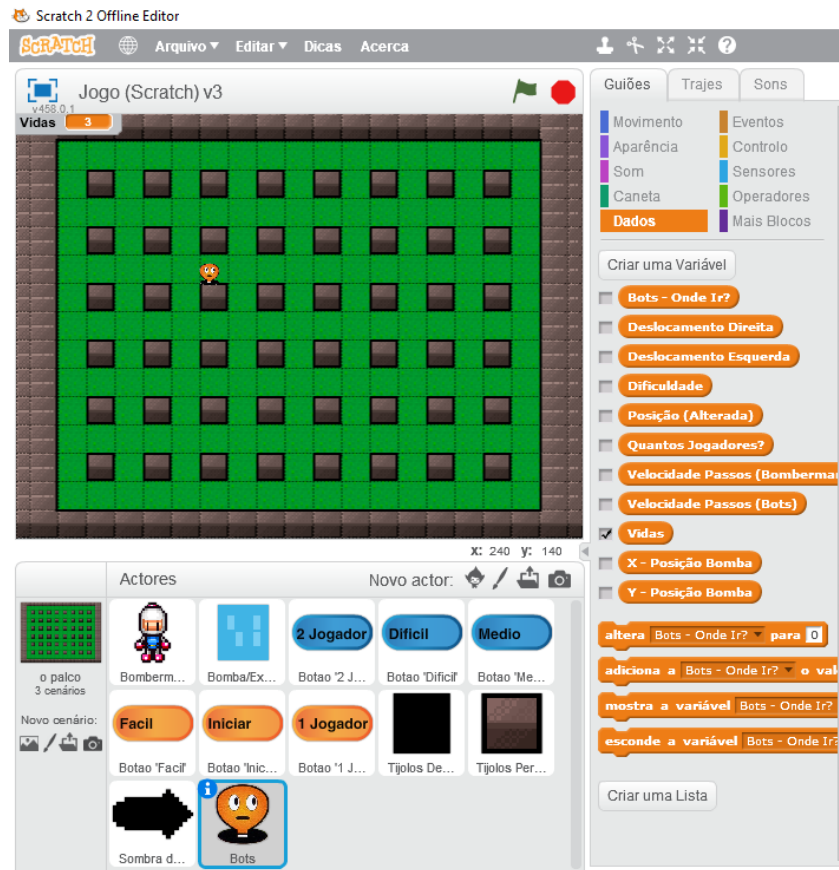
**Figura 24 - Vista Menu Inicial**

O passo seguinte consiste em criar *Bot's*. Os *Bot's* (diminutivo de *Robot*) são gerados de maneira aleatória no mapa onde não estejam "tijolos permanentes" nem destrutíveis e estes mudam de posição quando atingem alguma coordenada já ocupada por um bloco. São destruídos quando são atingidos pela bomba e caso o jogador lhes "toque" é retirada uma "vida" ao mesmo (figura 24). A figura abaixo conjectura a base do movimento dos *Bot's* com um movimento padrão constante. A figura 25 simboliza o código do movimento do *Bot*.



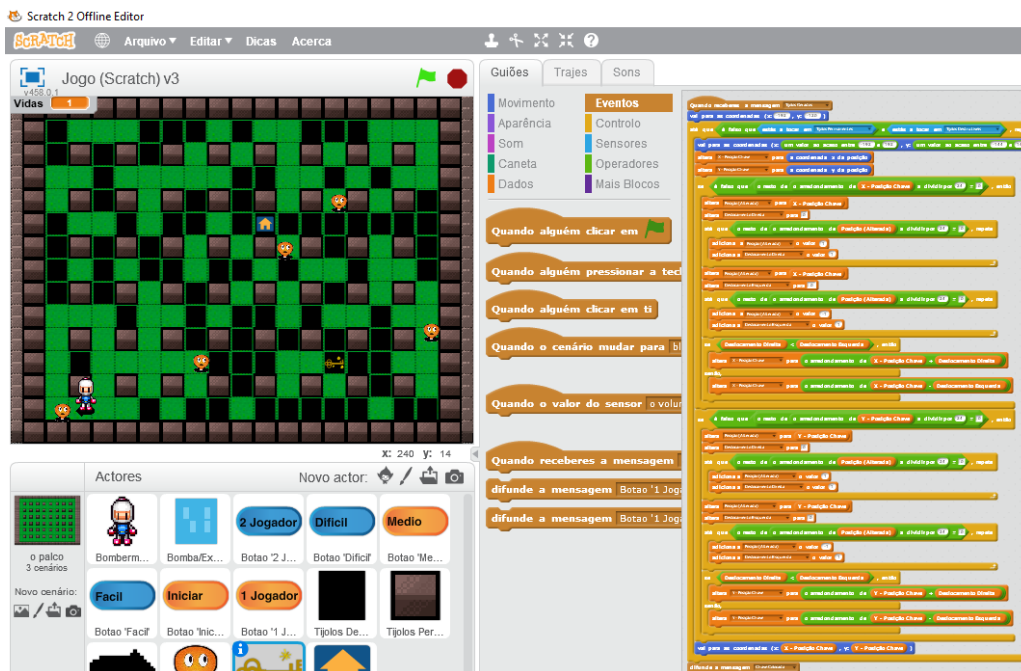
**Figura 25 - Algoritmo Movimento do Bot**





**Figura 26 - Vista Jogo com Bot**

De seguida foi trabalhada outra funcionalidade, de forma a tornar o jogo mais competitivo e divertido – a “Chave” e a “Porta”. Contêm mecânicas muito semelhantes ao da bomba, ou seja, é-lhes definida uma posição aleatória no mapa e fazem o mesmo ajuste de posição. No entanto, só interrompem o seu percurso quando procuram um tijolo destrutível e não tocam uma na outra. O jogador vence o jogo quando tiver obtido a “chave” e tiver encontrado a “porta”; mas se tentar entrar na “porta” sem ter obtido a “chave”, nada acontece (figura 26).



**Figura 27 - Vista de tela e algoritmo da "chave" e "porta"**

Implementou-se também outro modo que consiste em jogar com outra pessoa, sendo uma imitação exata do jogador 1, alterando apenas os controlos (teclas "WASD", em vez das setas direcionais e a tecla "Q" para plantar bomba, substituindo a tecla "Espaço"). O jogo também inclui menus de vitórias e derrotas e também a preferência do "Traje" para cada jogador.

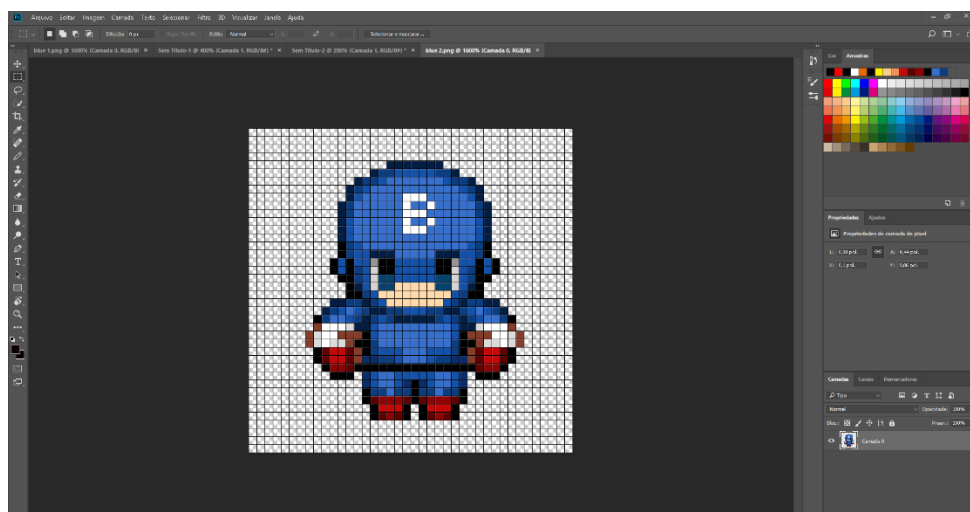
Questões a serem melhoradas no jogo: A velocidade a que a chave e a porta são escondidas, pois o tempo de espera é muito longo quando o utilizador deseja jogar neste modo; por vezes os *Bot's* conseguem sair dos limites do mapa. Desvantagens do "Scratch": pode ser demasiado lento, muitas das vezes; é possível a utilizar funções para um código mais compacto mas estas não funcionavam dentro do programa. Avaliação positiva do Scratch: é possível ver o valor das variáveis a qualquer momento, o que é muito bom para depurar erros que surjam no código.

### 3.1.6 Photoshop

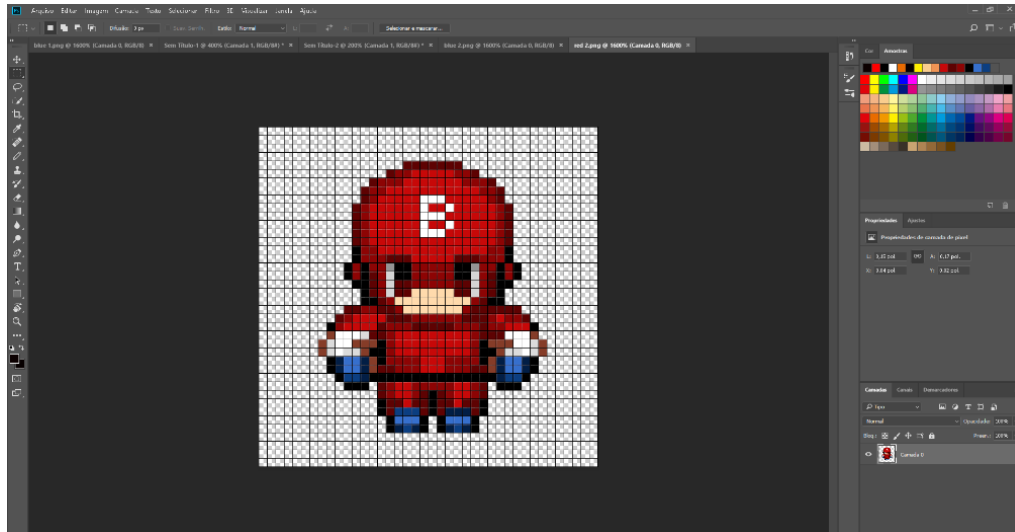
O software Adobe Photoshop, desenvolvido pela Adobe Systems, proporciona muitas ferramentas de edição de fotografias, criação de imagens, e design gráfico. É considerado o líder no mercado dos editores de imagem profissionais. Este programa foi utilizado com o intuito de criar novas personagens, objetos e fundos para o jogo, tais como os jogadores, os bots, os blocos de explosivos, a bomba, o rasto da explosão, a chave, a porta, o menu, etc.

O conjunto de imagens que se segue tenciona explicar os procedimentos realizados ao longo da criação de todos os objetos referidos no parágrafo anterior.

Nas figuras 22 e 23, estão representadas as duas personagens do jogo, "Bomberman" azul e "Bomberman" vermelho, respetivamente. Foram criados e inspirados no Capitão América, mas com algumas diferenças, como por exemplo a letra "B" na cabeça, que representa a inicial de "Bomberman". Começou-se por desenhar a silhueta dos corpos e dos pés. A primeira imagem que se conseguiu obter foi um boneco azul, só com uma pose, mas logo de seguida desenhou-se várias posições do mesmo boneco, dando a ideia de movimento.

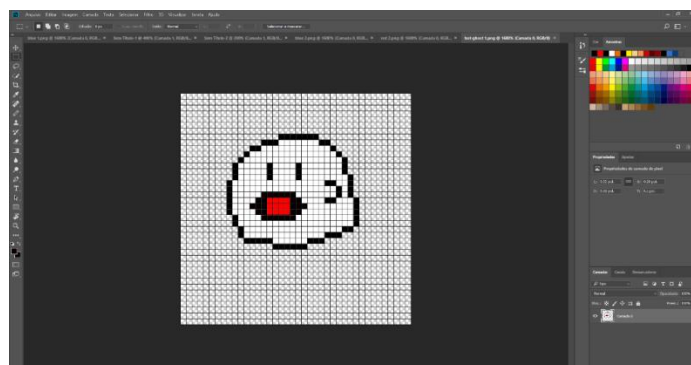


**Figura 28 - Criação do "Bomberman" azul**



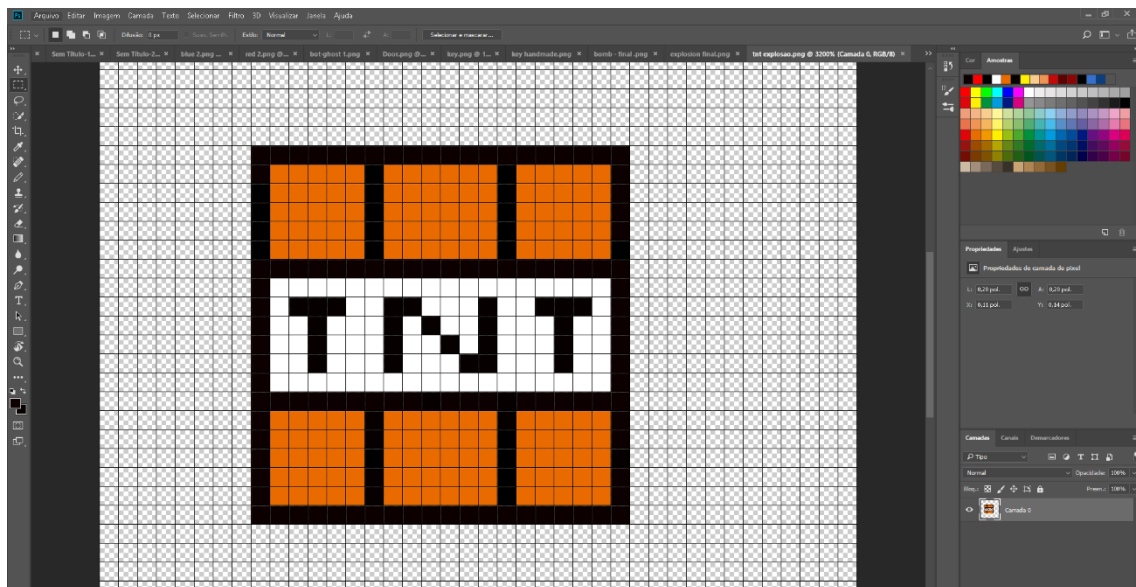
**Figura 29 - Criação do "Bomberman" vermelho**

Tal como foi feito nos personagens principais, os "NPC's" – "*Non-playable Characters*" (figura 24) foram criados a partir de uma silhueta de um fantasma, tendo apenas duas visões, uma onde é visível o vulto e outra em que tal não é possível. Apesar de parecer simples, o *design* deste personagem revelou-se o oposto.



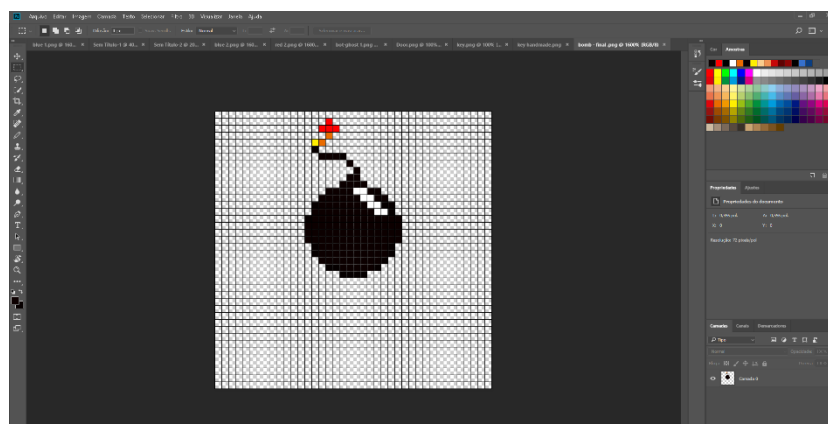
**Figura 30 - Criação dos Bot's**

O bloco explosivo (figura 25) foi retirado de um jogo chamado “Minecraft”. Como foi uma cópia completa, as alterações despenderam menos tempo, tendo, por isso, poucos detalhes. Existem dois blocos que, aleatórios, que são especiais, na medida em que cada um contém uma chave e uma porta, essenciais à vitória do jogo “Bomber Trap”.



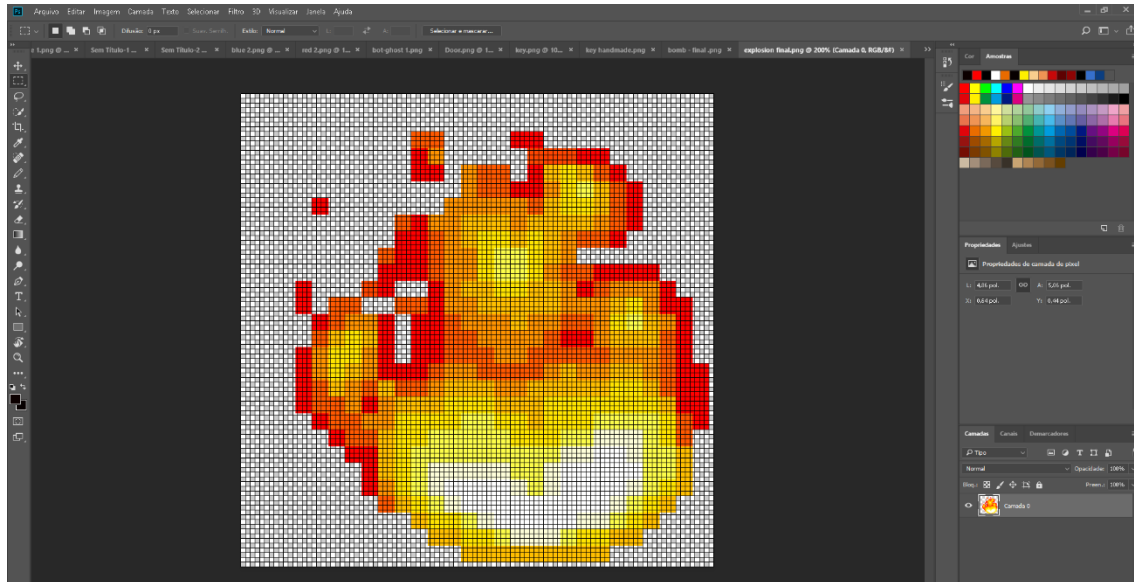
**Figura 31 - Bloco explosivo**

A figura ilustrativa da bomba (Figura 26) é o item com mais relevância no jogo, porque sem ela e sem a explosão era impossível encontrar a chave e a porta de saída do jogo para alcançar a vitória. O *design* dela foi inspirado nas balas dos canhões dos navios de piratas.



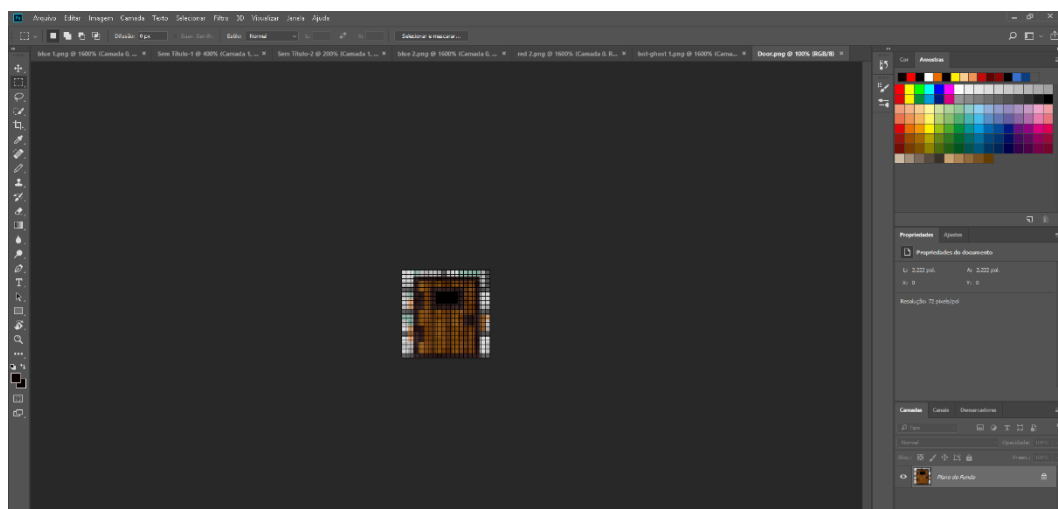
**Figura 32 - Criação da bomba**

A explosão (Figura 27) foi o item mais difícil de fazer e mais preocupante, no que toca ao tempo despendido. Como se pode reparar, a imagem tem muitos detalhes o que fez com que fosse necessário refazer diversas vezes.

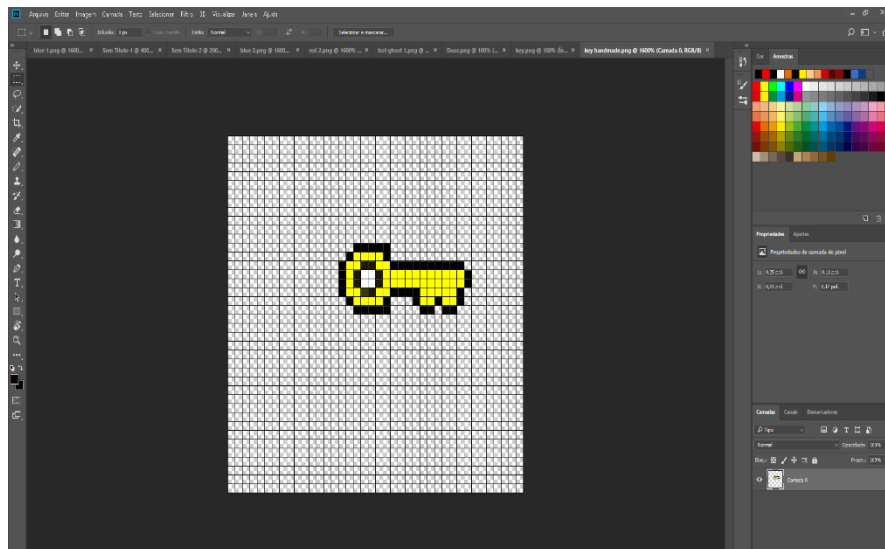


**Figura 33 - Criação da explosão**

A porta (Figura 28) e a chave (Figura 29), são cruciais para completar o jogo. Caso não sejam descobertas o jogo não termina.



**Figura 34 - Criação da porta**



**Figura 35 - Criação da chave**



**Figura 36 - Versão final do jogo**



## 4. Análise dos resultados

Na tabela abaixo (Tabela 3), estão indicados os requisitos que foram cumpridos (representados com o símbolo "+") e os que não foram cumpridos (representados com o símbolo "-").

**Tabela 3 – Análise do cumprimento dos requisitos; RF – requisito funcional; RNF – Requisito não funcional**

Id	Requisitos	Análise dos resultados
<b>RF1</b>	Menu inicial do jogo (escolher nº de jogadores, dificuldade, iniciar)	+
<b>RF2</b>	Os bonecos devem ser capazes de se mover para todos os lados	+
<b>RF3</b>	Existem bots que se movem de maneira aleatória pelo mapa	+
<b>RF4</b>	Existem partes do mapa que podem ser destruídas e outras não	+
<b>RF5</b>	Ao clicar em certa tecla o boneco tem de largar uma bomba	+
<b>RF6</b>	A bomba tem que explodir	+
<b>RF7</b>	A explosão não pode destruir partes do mapa	+
<b>RF8</b>	A explosão destrói bots e o boneco	+
<b>RF9</b>	O jogo acaba quando o boneco for destruído ou se destruir todos os bots	+
<b>RF10</b>	Existir vários níveis de dificuldade em que se aumenta a velocidade dos bots, da explosão e o número de bots	_*
<b>RF11</b>	Destruindo todos os bots passa-se de nível	-
<b>RNF1</b>	Menu de configurações (tirar ou não o som da musica e os efeitos sonoros)	-
<b>RNF2</b>	Menu de pausa	-
<b>RNF3</b>	Quando sair do menu de pausa dar 3 segundos antes de começar	-
<b>RNF4</b>	Jogo pode correr em <i>full screen</i>	+

\*Apesar de existirem 3 níveis de dificuldade (Fácil, Médio e Difícil), apenas o número de bots aumenta, ao contrário da velocidade dos mesmos e o detonar da bombar.



## 5. Conclusão

No decorrer do presente relatório, pretendeu-se descrever a estrutura do projeto durante o seu desenvolvimento, bem como das etapas inerentes à criação do jogo. No mapa de Gantt previsto, como o próprio nome indica, prevê-se o rumo do projeto, não se tendo uma perceção total do tipo de imprevistos que podem ocorrer ao longo da realização do mesmo. As tarefas têm uma duração estimada, mas no fim acabam por durar, geralmente, mais tempo, como aconteceu com o desenvolvimento do jogo. Assim, o mapa de Gantt inicial serve essencialmente para organizar as tarefas e a sua distribuição, pois a duração não é certa. Segundo o mapa de Gantt executado e já atualizado, consideramos que se conseguiu cumprir as tarefas e os requisitos, no geral, propostos.

Relativamente ao software de gestão de projetos, Microsoft Project, este é de fácil manuseamento, permitindo ao utilizador organizar as suas tarefas de acordo com as suas relações de precedência e durações (colocando data de começo e finalização da tarefa). No caso de projetos com múltiplas tarefas, ajuda na observação da evolução de cada uma delas e consequente evolução do projeto, podendo serem efetuados ajustes à medida que determinados problemas venham a ocorrer.

O software utilizado na criação do jogo, Scratch, revelou-se, por vezes, demasiado lento e, apesar de ser possível o uso de funções para tornar o código mais “limpo”, quando se tentava executar não funcionava. No entanto, o facto de o valor das variáveis ser visível a qualquer momento, permite que se descubra o porquê de o código não estar a ser executado como o suposto. Concluiu-se também que a velocidade a que a chave e a porta são escondidas podia ser melhorada, pois atualmente o tempo de espera é muito longo. Reparámos também que, muito excecionalmente, alguns bots conseguiam sair dos limites do mapa, um erro que não foi totalmente corrigido.

Em relação ao Photoshop, utilizado para editar/criar as imagens requeridas pelo jogo, mostrou-se, num primeiro contacto, um software não muito acessível, apesar de todas ferramentas que disponibiliza. No entanto,

com a prática e à medida que se despendia mais tempo no mesmo, acabou por se tornar mais eficiente.

Quanto à plataforma disponibilizada pela Universidade de Aveiro, Code UA, esta foi maioritariamente utilizada para atribuir tarefas e prazos a cada membro do grupo, tendo sido bastante útil a visualização do estado em que se encontrava cada uma (% Completo). Permitiu também a troca de ideias e informação entre o grupo e a Professora Orientadora. Quanto ao mapa de Gantt preferimos o fornecido pelo software MS Project, por uma questão de hábito e facilidade de manipulação. Já o repositório (Git), permitiu a atualização de cada documento e respetiva partilha entre os membros do grupo, tendo sido possível a adição de novas versões sem que se perdessem a informação das anteriores.

Apesar de não se ter conseguido pôr em prática todos os requisitos previamente definidos, como o aumento da velocidade dos bots e da explosão, a passagem de nível através da destruição de todos os bots, e a existência de um menu de configurações e de um menu de pausa, foram também adicionadas outras funcionalidades que não constavam inicialmente, como é o caso da introdução da chave e da porta – objetos dispostos aleatoriamente, graças ao código – e da vitalidade de cada personagem. Caso se esteja a jogar no modo “2 jogadores”, independentemente da personagem, este número condiciona a finalização do jogo. O mesmo acontece em “1 jogador”. À medida que se avança em dificuldade, a vitalidade inicial diminui.

Deste modo, é sempre importante e uma mais-valia poder elaborar algum projeto em grupo, pois providencia à discussão de diferentes ideias e opiniões, a cooperação, trabalho mútuo, mais confiança na tomada de decisão e a propensão de risco é menor. Todos esses aspetos conjugados com o desenvolvimento de um projeto inserido numa das áreas mais ricas e com mais oportunidades nos dias de hoje.

Concluindo, com o constante avanço dos meios tecnológicos e dos recursos que vão surgindo, exige-se cada vez mais técnicos e especialistas

na área das novas tecnologias e da informação. Com efeito, adquirir competências no trabalho em equipa, saber gerir um trabalho com outros colegas, desenvolver a capacidade de comunicação são qualidades que muitas empresas procuram muito numa pessoa. As empresas procuram pessoas que saibam resolver problemas, e não o contrário. Arranjar problemas qualquer um o faz, pois tal como disse Ralph Waldo Emerson: “um problema sem solução é um problema mal colocado”.

## 6. Bibliografia

Emerson, R. W. (s.d.). *Um problema sem solução é um problema mal colocado*. . Obtido de Citador: <http://www.citador.pt/frases/um-problema-sem-solucao-e-um-problema-mal-colocado-ralph-waldo-emerson-6041>

Microsoft. (s.d.). *Gestão de Projetos da Microsoft*. Obtido de Microsoft: <https://products.office.com/pt-pt/project/project-management>

Priberam Informática, S.A. (s.d.). *Requisito*. Obtido de Priberam Dicionário: <https://www.priberam.pt/dlpo/requisito>

Software Freedom Conservancy. (s.d.). *Primeiros passos - Noções Básicas de Git*. Obtido de GitHub: <https://git-scm.com/book/pt-br/v1/Primeiros-passos-No%C3%A7%C3%B5es-B%C3%A1sicas-de-Git>