



Instituto Politécnico de Viana do Castelo

Escola Superior
de Tecnologia
e Gestão



MONOPÓLIO

PROJETO FINAL

PROJETO FINAL
PROJETO 2D

DOCENTES: LUÍS ROMERO

DIOGO MIGUEL ALMEIDA AMORIM Nº:18463
TIAGO ANDRÉ CAMPOS CASTRO Nº:18459

1. Breve descrição do jogo

O MONOPOLY é o jogo onde se compram, alugam ou vendem propriedades de uma maneira vantajosa, de forma a que os jogadores aumentem o seu capital – sendo o vencedor o jogador mais rico. Começar na casa PARTIDA e deslocar os peões à volta do tabuleiro, de acordo com a pontuação obtida nos dados. Quando o peão de um jogador parar num espaço sem proprietário, ele pode decidir comprá-lo ao Banco. Se ele(a) decidir não comprar o espaço, este é vendido em leilão pela melhor oferta. Os jogadores que têm propriedades recebem rendas dos adversários que aí pararem. A construção de Casas e de Hotéis aumenta consideravelmente as rendas pagas pelas propriedades, logo, é conveniente construir no máximo de terrenos possível. Com o objetivo de conseguir mais dinheiro, os Terrenos de Construção podem ser hipotecados pelo Banco. Os jogadores devem seguir sempre as instruções das cartas da Caixa da Comunidade e da Sorte. Por vezes os jogadores vão para a Cadeia.

1.1 Objetivo

Ser o último jogador no tabuleiro que não foi à falência.

1.2 Equipamento

- 1 tabuleiro de jogo
- 10 peões
- 28 cartas de títulos de propriedade
- 16 cartas da sorte
- 16 cartas da caixa da comunidade
- Dinheiro do MONOPOLY
- 32 casas
- 12 hotéis
- 2 dados

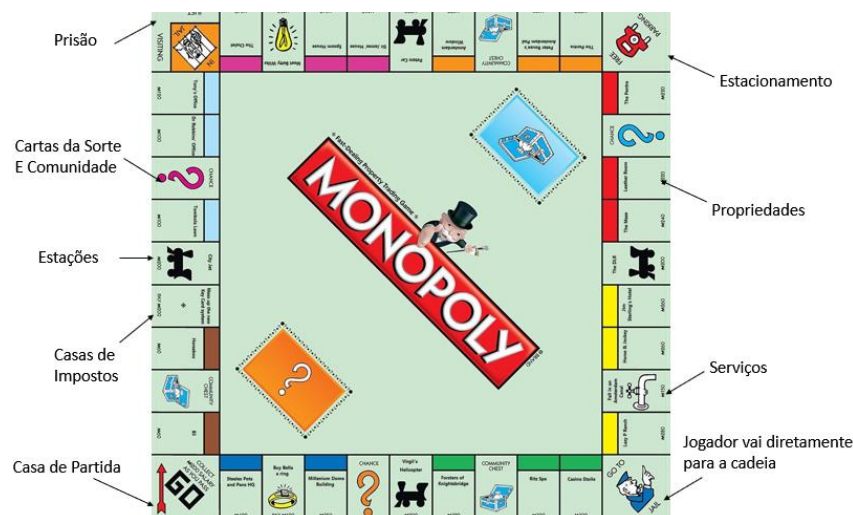


Figura 1- Tabuleiro com identificação das casas

1.3 Preparação

1. Colocar as Casas, Hotéis, Títulos de Propriedade e dinheiro (por ordem de valor) nas diversas secções do tabuleiro de jogo.
2. Separar as cartas da sorte e as cartas da caixa da comunidade, baralhar e colocar nos respetivos lugares do tabuleiro de jogo.
Colocar os suportes das cartas nos seus lugares no tabuleiro do jogo.
3. Separar as Cartas da Sorte e as cartas da comunidade, baralhar e colocá-las nos suportes viradas para baixo.
4. Cada jogador escolhe uma peça e coloca-a na casa de PARTIDA.
5. O Banqueiro e a Banco - Um dos jogadores é escolhido para Banqueiro. Se existirem mais do que cinco jogadores, o Banqueiro pode escolher assumir apenas este papel.
O Banqueiro dá a cada jogador 150.000€, distribuídos da seguinte forma:
 - 2 notas de 50.000€
 - 4 notas de 10.000€
 - 1 nota de 5.000€
 - 1 nota de 2.000€
 - 2 notas de 1.000€
 - 1 nota de 500€
 - 5 notas de 100€

Além de conservar o dinheiro restante, o Banco conserva também as cartas de títulos de propriedade, as casas e os hotéis até eles serem comprados pelos jogadores. O Banco paga salários e bonificações, empresta dinheiro sobre hipotecas e recebe todos os impostos, multas, empréstimos e juros. Em caso de leilão, o Banqueiro funciona como leiloeiro.

O Banco nunca vai à falência, mas pode emitir todo o dinheiro necessário sobre a forma de títulos de dívida escritos em papel normal.

6. Os jogadores lançam ambos os dados. O jogador que tiver obtido a pontuação mais elevada é o primeiro a jogar e o jogo prossegue pelo lado esquerdo.

1.4 O Jogo

Na sua vez de jogar, lançar os dois dados e avançar na direção da seta. A casa a que chegar indica o que deve fazer. Podem existir duas ou mais peças na mesma casa ao mesmo tempo. Conforme a casa em que parar, pode ter de seguir uma das seguintes instruções:

- Comprar terrenos de construção ou outras propriedades
- Pagar renda se a propriedade pertencer a outro jogador
- Pagar impostos
- Tirar uma carta da sorte ou da caixa da comunidade
- Ir para a cadeia
- Ficar no estacionamento livre
- Receber 20.000€

1.5 Doubles

Se um jogador, ao lançar os dados, obtiver um *double*, desloca o peão como habitualmente e age de acordo com as regalias ou penalidades dessa casa. Volta a lançar os dados novamente e desloca-se outra vez. Se lhe sair um *double* pela terceira vez consecutiva, deve ir imediatamente para a cadeia.

1.6 Partida

Sempre que um peão de um jogador parar ou passar na casa de PARTIDA, enquanto se desloca na direção da seta, recebe do Banqueiro uma bonificação no valor de 20.000€. É possível receber 20.000€ duas vezes na mesma jogada se, por exemplo, para na casa da sorte ou na casa da caixa da comunidade imediatamente após a PARTIDA e lhe sair a carta que lhe indica para avançar para a casa de PARTIDA.

1.7 Comprar Propriedades

Quando um jogador parar sobre um terreno sem proprietário (isto é, sobre um terreno de construção do qual nenhum jogador detenha o título de propriedade), pode optar por comprar essa propriedade ao Banco pelo preço indicado na respetiva casa. Se o jogador opta por comprar, paga a propriedade ao Banco e recebe como prova de compra o título de propriedade. Deve colocá-lo à sua frente, virado para cima. Se o jogador opta por não comprar, o Banqueiro deve imediatamente oferecer esta propriedade para venda em leilão pela melhor oferta. O preço de partida de partida para o leilão pode ser qualquer preço que outro jogador estiver disposto a oferecer. Todos os jogadores podem participar no leilão, incluindo aqueles que optaram por não comprar pelo preço original.

1.8 Ser Proprietário

Ser proprietário permite a um jogador receber de quaisquer “inquilinos” que pararem nessa casa. É vantajoso possuir os títulos de propriedade de todos os terrenos de um Grupo de Cor – por outras palavras, possuir um monopólio. Um jogador pode construir em quaisquer terrenos dos quais possua o grupo de cor completo.

1.9 Paragem sobre um terreno com Proprietário

Quando um jogador parar um terreno previamente comprado por outro jogador, pode ser necessário pagar renda para poder parar. O jogador que possui este terreno tem de cobrar a renda antes de se voltarem a lançar os dados. A quantia a pagar está indicada no respetivo título de propriedade e varia de acordo com o número de edifícios construídos. Se todas as propriedades de um grupo de cor estiverem na posse de um jogador, o proprietário cobra a dobrar em qualquer terreno que ainda não possua construções. No entanto o proprietário que possui um grupo de cor completo não pode cobrar renda a dobrar se algum dos edifícios estiver hipotecado. Se o terreno contém uma ou mais casas, a renda é mais alta conforme o indicado no respetivo título de propriedade. Não se pagam rendas de terrenos hipotecados.

1.10 Paragem em casas de Serviços

Se um jogador parar numa destas casas, pode comprar o serviço se ainda o não possuir. Como acontece com as outras casas de propriedade, deve pagar ao Banco o preço indicado nessa casa. Se a propriedade já tiver proprietário, pode ter que pagar renda de acordo com o número que obteve no lançamento dos dados. Se o proprietário só possuir um dos serviços a renda será quatro vezes superior ao número obtido através do lançamento dos dados. Se, por outro lado, ambos os serviços pertencerem ao mesmo jogador, terá de pagar dez vezes a quantia indicada no lançamento dos dados. Se o jogador chegou aqui através de uma carta da sorte ou da comunidade, deve lançar os dados para saber quanto é que deve pagar.

1.11 Paragem nas Estações

Se for o primeiro jogador a chegar tem oportunidade de comprar a Estação. De outro modo ela será leiloadada pelo Banco, e, mesmo que tenha optado por não comprar pelo preço original, poderá participar. Se a estação já tiver sido adquirida quando chegar, deverá pagar a quantia indicada no título de propriedade. A quantia a pagar varia consoante o número de outras estações possuídas por esse jogador.

1.12 Paragem na “Sorte” ou na “Caixa da Comunidade”

Os jogadores que pararem nestas casas tiram a carta que se encontra no topo do respetivo baralho. Estas cartas podem pedir-lhe para:

- Deslocar a sua peça
- Pagar dinheiro – por exemplo, em impostos.
- Receber dinheiro.
- Ir para a cadeia
- Sair da cadeia

Deve seguir as instruções indicadas na carta e atuar imediatamente antes de voltar a colocar a carta no fundo do baralho. Os jogadores podem guardar a carta “Você está livre da cadeia” até precisarem dela ou vendê-la a outro jogador por um preço acordado mutuamente.

Nota: Uma carta pode dizer-lhe para deslocar a sua peça para outra casa. Se, no caminho, passar pela casa de PARTIDA, receberá 20.000€. Não passa pela casa de PARTIDA quando é enviado para a prisão.

1.13 Paragem em Casas de Impostos

Quando parar numa casa de impostos deve pagar todos os impostos à Banca.

1.14 Paragem em Estacionamento Livre

Quando parar numa destas casas, fica nela até à próxima jogada. Não existe qualquer penalidade, poderá continuar a fazer transações como habitualmente (por exemplo, receber rendas, construir em terrenos que possua, etc.)

1.15 Cadeia

Um jogador vai para a cadeia quando:

- O seu peão parar na casa “VÁ PARA A CADEIA”;
- Sair uma carta da sorte ou da caixa da comunidade que lhe diz para ir “DIRECTAMENTE PARA A CADEIA”;
- Obter *doubles* três vezes consecutivas na sua jogada;

Quando um jogador é envidado para a cadeia, a sua vez acaba imediatamente. Neste caso, não recebe os 20.000€ de bonificação, independentemente da casa do tabuleiro em que estiver.

Para sair da cadeia, o jogador pode:

- Pagar uma multa de 5.000€ e continuar na próxima jogada;
- Comprar uma carta “Você está livre da cadeia” a outro jogador, por um preço acordado mutuamente e usá-la para sair da cadeia;
- Esperar na cadeia durante três jogadas, lançando os dados em cada uma delas para tentar obter *double*. Se obtiver *double* pode sair da cadeia e avançar o número de casas indicadas;

Enquanto estiver na cadeia pode receber rendas desde que as propriedades não estejam hipotecadas. Se o jogador não for enviado para a cadeia, mas durante o jogo parar nessa casa, é apenas um visitante e não incorre em qualquer penalidade. Na jogada seguinte, avança como usualmente.

1.16 Casas

Quando um jogador possui todos os terrenos de um grupo de cor pode comprar casas para colocar em qualquer uma desses espaços. Isto aumenta a renda que pode cobrar aos adversários que passarem por estes terrenos. O preço de uma casa está indicado no respetivo título de propriedade. O jogador pode comprar na sua vez de jogar ou entre as jogadas de outros jogadores, mas tem de construir de uma forma uniforme, isto é, não pode construir mais de uma casa em cada terreno de um grupo de cor até que tenha construído uma casa em todos os terrenos desse grupo, e assim consecutivamente até atingir o limite de quatro casas em cada terreno. Os terrenos também devem ser vendidos de uma forma uniforme. Pode vender e comprar casas em qualquer altura de acordo com o que a sua situação financeira permitir. As casas não podem ser construídas se qualquer dos terrenos do grupo de cor estiver hipotecado. Se o jogador possuir todos os terrenos de um grupo e cor e só um ou dois tiverem construções, pode continuar a receber renda a dobrar de qualquer jogador que pare num dos terrenos sem construções desse grupo de cor.

1.17 Hotéis

Os jogadores devem ter quatro casas em cada terreno de um grupo de cor completo para poderem comprar um Hotel. Os hotéis são comprados como as casas, e custam o preço de quatro casas, que são devolvidas ao Banco, em conjunto com o preço indicado no título de propriedade. Só se pode construir um hotel em cada terreno.

1.18 Falta de construções

Se o banco não tem mais casas para venda, os jogadores que quiserem construir têm que esperar que outro jogador devolva ou venda casas ao banco para o poderem fazer. Da mesma forma, quando pretende vender hotéis não pode substituí-los por casas se já não existirem casas disponíveis.

Se há um número limitado de casas ou hotéis disponíveis e dois ou mais jogadores quiserem comprar um número superior ao que a banca dispõe, esta deve vender as casas e hotéis pela melhor oferta, começando pelo preço mais baixo indicado no título(s) de propriedade relevante(s).

1.19 Venda de propriedades

Os terrenos sem construções, as Estações de caminho de ferro e as Empresas de Serviço Público podem ser vendidas a qualquer jogador como se tratasse de uma operação particular e mediante uma quantia conveniente a ambos. No entanto nenhum terreno pode ser vendido a outro jogador se existir alguma construção em qualquer dos terrenos desse grupo de cor. Se algum proprietário desejar vender um terreno pertencente a um grupo de cor, devolve ao banco todos os edifícios construídos nesse terreno.

As casas têm de ser vendidas de uma maneira uniforme (ver Casas).

As casas e os hotéis não podem ser vendidos a outros jogadores. Têm de ser vendidos ao banco por metade do valor indicado no respetivo título de propriedade. Podem ser vendidos em qualquer altura.

No caso dos hotéis, a banca pagará metade do seu preço de compra mais metade do preço das quatro casas que foram dadas à banca pela aquisição do hotel.

1.20 Hipotecas

Se um jogador já não tiver dinheiro e precisar de pagar uma dívida, pode obtê-lo através da hipoteca de uma propriedade. Para isso, começa por vender todos os terrenos existentes nessa propriedade à banca. Para hipotecar uma propriedade, virar o respetivo título de propriedade ao contrário e receber da banca a hipoteca no valor indicado nas costas. Quando desejar pagar a hipoteca deve pagar esta quantia acrescida de 10% de juros. Se hipotecar uma propriedade, continua a deter a sua posse. Nenhum outro jogador a pode adquirir pagando a hipoteca ao banco. O proprietário não pode receber renda de uma propriedade hipotecada apesar de poder continuar a receber rendas de outras propriedades desse grupo de cor. O jogador pode vender propriedades hipotecadas a outros jogadores por um preço acordado mutuamente. O comprador pode decidir retirar imediatamente a hipoteca pagando a hipoteca e 10% de juros. Em alternativa, pode pagar os 10%, mas manter a hipoteca. Neste caso, quando a hipoteca for finalmente retirada tem de pagar novamente 10% de juros. Quando todos os terrenos de um grupo de cor já não estiverem hipotecados, o proprietário pode começar a comprar casas pelo seu preço total.

1.21 Falência

Se um jogador dever ao banco ou a outro jogador mais do que pode pagar abre falência e retira-se do jogo. Se a dívida for ao banco, entrega-lhe todo o seu dinheiro e títulos de propriedade. O banco em seguida vende-os em leilão pela melhor oferta. O jogador falido deve colocar a carta “sair livre da cadeia” no fundo do baralho respetivo. Se a dívida for a outro jogador, as casas e os hotéis são vendidas ao banco por metade do preço original e o credor recebe todo o dinheiro, títulos de propriedade e cartas “está livre da cadeia” que o jogador possuir. Se possuir propriedades hipotecadas, deve entregá-las e o credor pagará imediatamente 10% de juros e decidirá se mantém a hipoteca ou não.

1.22 Notas sobre o jogo

Se um jogador não tem dinheiro suficiente para pagar as suas rendas, pode pagar ao seu credor uma parte em dinheiro e outra parte em propriedades. Neste caso, o credor pode optar por aceitar certas propriedades, (mesmo hipotecadas), por um valor superior ao indicado, de maneira a obter mais propriedades para construção ou para evitar que outro jogador fique em poder dessa propriedade.

Os proprietários dos terrenos devem ter atenção às rendas das suas propriedades.

O banco empresta dinheiro apenas contra hipotecas.

Os jogadores não podem emprestar dinheiro ou propriedades entre si.

1.23 Ganhar

O último jogador a permanecer em jogo é o vencedor.

1.24 CARTAS DA SORTE

- Você está livre da cadeia;
(Esta carta pode ser conservada até ser necessária, ou vendida)
- O banco paga-lhe dividendos no valor de 5.000€;
- Receba o seu empréstimo para construção, 15.000€;
- Faça reparações em todas as suas casas:
 - Pague, por cada casa → 2.500€;
 - Pague, por cada hotel → 10.000€;
- Recue três casas;
- Pague por despesas escolares, 15.000€;
- Vá para a cadeia;
(Vá diretamente para a cadeia, sem passar na casa da partida e sem receber 20.000€)
- Avance até à Rua Júlio Diniz (Porto);
(Se passar na casa PARTIDA receba 20.000€)
- Multa por embriaguez. Pague 20.000€;
- Avance até à Av. Central (Braga);
(Se passar pela casa PARTIDA receba 20.000€)
- Avance até ao Rossio (LISBOA). Recebe 20.000€;
- Você ganhou o concurso das palavras cruzadas. Receba 10.000€;
- Avance até à casa PARTIDA;
- Vá até à estação de Campanhã;
(Se passar na casa PARTIDA receba 20.000€)
- Você tem de pagar, para reparações da rua:
 - Por casa → 4.000€;
 - Por Hotel → 11.500€;
- Multa, por excesso de velocidade, 1.500€;

1.25 CAIXA DA COMUNIDADE

- Você herdou 10.000€;
- Pague a conta do médico, 5.000€;
- Erro da banca a seu favor. Receba 20.000€;
- Receba, por reembolso do imposto sobre capitais, 2.000€;
- Pague uma multa de 1.000€ ou, tire uma carta da sorte;
- Pague o seu seguro, 5.000€;
- Avance até à casa PARTIDA;
- Receba a sua renda anual, 10.000€;
- É o seu aniversário. Receba 1.000€ de cada jogador;
- Você ficou em segundo lugar num concurso de beleza, receba 1.000€;
- Receba, pela venda do seu stock, 5.000€;
- Receba 7% de juros sobre as suas ações, 2.500€;
- Pague a conta do hospital, 10.000€;
- Vá para a cadeia;
(Vá diretamente para a cadeia, sem passar na casa de partida e sem receber 20.000€)
- Recue até ao campo grande (Lisboa);
- Você está livre da cadeia;
(Esta carta pode ser conservada até ser necessária, ou vendida)

1.26 TÍTULOS DOS SERVIÇOS

Se um jogador possui uma EMPRESA DE UTILIDADE PÚBLICA, o aluguer será de 400€ vezes o número indicado nos dados.

Se um jogador possui duas EMPRESAS DE UTILIDADE PÚBLICA, o aluguer será de 1.000€ vezes o número indicado nos dados.

VALOR DE HIPOTECA: 7.500€

1.27 TÍTULOS DAS ESTAÇÕES

ALUGUER → 2.500€

Se tem 2 estações → 5.000€

Se tem 3 estações → 10.000€

Se tem 4 estações → 20.000€

Valor de Hipoteca → 10.000€

1.28 TÍTULOS DE PROPRIEDADES

EXEMPLO:

- Campo Grande (Lisboa):
 - Aluguer Só do terreno → 200€;
 - Aluguer Com 1 Casa → 1.000€;
 - Aluguer Com 2 Casas → 3.000€;
 - Aluguer Com 3 Casas → 9.000€;
 - Aluguer Com 4 Casas → 16.000€;
 - Aluguer Com HOTEL → 25.000€;

Preço das Casas → 5.000€ Cada

Preço do Hotel → 5.000€ mais 4 casas

Valor de Hipoteca → 3.000€

2. Ambiente de Programação



Para programar este jogo necessitava-mos de alguns requisitos:

- Possibilidade de criação de gráficos (GUI);
- Permitir animação;
- Manipulação de eventos;

Os Ambientes disponíveis para realização deste tipo de tarefas eram:

- Visual C#;
- Visual Basic;
- Java/Java Swing – recorrendo ao IDE NetBeans;

Com isto, tivemos a nossa escolha entre C# e Java, então fomos á procura de quais as vantagens e desvantagens de cada um:

	
VANTAGENS	VANTAGENS
<ul style="list-style-type: none"> • Maior interatividade com os internautas; • Possui recurso de programação orientada a objeto; • Desenvolvimento em ambiente Windows; • Não precisa registrar componentes; • Possui fácil validação de dados; • Fácil tratamento de erros; 	<ul style="list-style-type: none"> • Arquitetura robusta e bem; • Facilita no desenvolvimento em equipe; • Facilita encontrar erros mais cedo, já que as variáveis são fortemente tipadas e isso facilita que os compiladores e IDEs mostrem os erros antes da execução; • Diversas implementações de servidores de aplicação e inúmeros frameworks web e desktop; • Permite a criação de apps desktop multiplataforma; • A Orientação a Objetos do Java é consistente;
DESVANTAGENS	DESVANTAGENS
<ul style="list-style-type: none"> • Os programas e componentes antigos devem ser reescritos; • Dífícil aprendizagem para programadores mais ortodoxos; • Não é uma linguagem de médio nível; • Não disponível em todas as plataformas; 	<ul style="list-style-type: none"> • Performance reduzida: por ser uma linguagem não projetada para gerar código nativo para cada arquitetura, os programas em Java costumam ter um desempenho menor que os escritos em linguagens como C++. Eles também costumam ocupar mais espaço em memória. Mas à medida que a linguagem vem sendo refinada, tal desvantagem vem diminuindo cada vez mais. • Programas em Java geralmente demoram mais tempo para compilar. • Programas que realizam cálculo com números em ponto flutuante tendem a possuir baixo desempenho.

3. Desenvolvimento e Construção do Jogo

Ao desenvolver este jogo vimo-nos com algumas dificuldades, nomeadamente a estruturação do mesmo. A construção do método lógico foi complexa, mas superada.

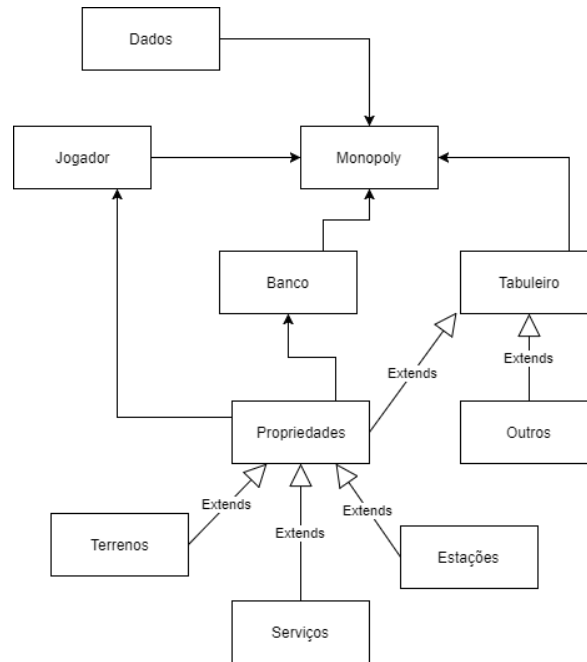


Figura 2- Diagrama de Classes Inicial

Na figura acima podemos ver o diagrama de classes que originou e potenciou este nosso programa, ou seja, toda a organização do jogo dentro do software a ser desenvolvido.

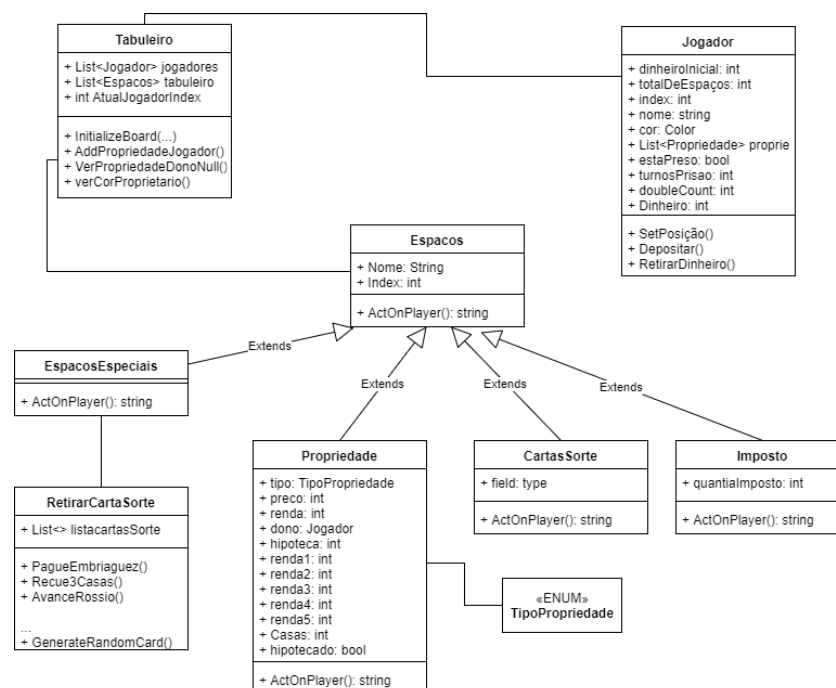


Figura 3-Diagrama de classes Final

3.1 Classes

- Espaços

Esta classe é onde são desenhados os espaços do tabuleiro e alocados o nome do jogador onde este se encontra.

```

namespace Monopoly.Classes
{
    7 referências
    public abstract class Espacos
    {
        4 referências
        public Espacos(int index, string nome)
        {
            this.Index = index;
            this.Nome = nome;
        }

        29 referências
        public int Index { get; private set; }

        24 referências
        public string Nome { get; private set; }

        12 referências
        public abstract string ActOnPlayer(Jogador jogador);
    }
}

```

Figura 4- Onde são colocadas o espaço das cartas da sorte no Tabuleiro

- Tabuleiro

Esta classe podemos dizer que é a mais importante do jogo onde são alocados todos os jogadores, ou seja sejam eles 2 ou 8, e todas as cartas dos respectivos tipos, Propriedades, Cartas da Sorte, Cartas Especiais e Impostos.

```

39 referências
public static class Tabuleiro
{
    public static List<Jogador> jogadores;
    public static List<Espacos> tabuleiro;
    public static int AtualJogadorIndex;
    public static int NrTotalJogadores;
    public static int NrRondas;

    1 referência
    public static void InitializeBoard(int NrJogadores, string nome1, string nome2, string nome3, string nome4, string nome5, string nome6, string nome7, string nome8, C
    {
        NrTotalJogadores = NrJogadores;
        AtualJogadorIndex = 0;
        NrRondas = 1;
        switch (NrJogadores)
        {
            case 2:
                jogadores = new List<Jogador>()
                {
                    new Jogador(1, nome1, cor1, peça1),
                    new Jogador(2, nome2, cor2, peça2),
                };
                break;
            case 3:
                jogadores = new List<Jogador>()
                {
                    new Jogador(1, nome1, cor1, peça1),
                    new Jogador(2, nome2, cor2, peça2),
                    new Jogador(3, nome3, cor3, peça3),
                };
                break;
            case 4:
                jogadores = new List<Jogador>()
                {
                    new Jogador(1, nome1, cor1, peça1),
                    new Jogador(2, nome2, cor2, peça2),
                    new Jogador(3, nome3, cor3, peça3),
                    new Jogador(4, nome4, cor4, peça4),
                };
                break;
        }
    }
}

```

Figura 5- Alocação dos nomes, cores e peças de todos os jogadores

```

tabuleiro = new List<Espacos>()
{
    new EspacosEspeciais(0,"PARTIDA! Passando Aqui Receba 20.000€"),
    new Propriedade(1,"Campo Grande (Lisboa)",TipoPropriedade.Brown,6000, 200, 3000, 1000, 3000, 9000, 16000, 25000),
    new CartasSorte(2,"Caixa da Comunidade"),
    new Propriedade(3,"Av. Fernão de Magalhães (Porto)",TipoPropriedade.Brown,6000, 400, 3000, 1000, 3000, 9000, 16000, 25000),
    new Imposto(4,"Pague Imposto Sobre Capitais", 20000),
    new Propriedade(5,"Estação do Rossio",TipoPropriedade.Estação,20000, 2500, 10000, 0, 0, 0, 0, 0),
    new Propriedade(6,"Avenida Almirante Reis (Lisboa)",TipoPropriedade.Blue,10000, 600, 5000, 3000, 9000, 27000, 40000, 55000),
    new CartasSorte(7,"SORTE"),
    new Propriedade(8,"Avenida TODI (Setúbal)",TipoPropriedade.Blue,10000, 600, 5000, 3000, 9000, 27000, 40000, 55000),
    new Propriedade(9,"Av. 24 de Julho (Lisboa)",TipoPropriedade.Blue,12000, 800, 6000, 4000, 10000, 30000, 45000, 60000),
    new EspacosEspeciais(10,"Cadeia"),
    new Propriedade(11,"Av. Central (Braga)",TipoPropriedade.Pink,14000, 1000, 7000, 5000, 15000, 45000, 62500, 75000),
    new Propriedade(12,"Companhia Eletrecidade",TipoPropriedade.Companhia,15000, 400, 7500, 0, 0, 0, 0, 0),
    new Propriedade(13,"Rua Ferreira Borges (Coimbra)",TipoPropriedade.Pink, 14000, 1000, 7000, 5000, 15000, 45000, 62500, 75000),
    new Propriedade(14,"Av. de Roma (Lisboa)",TipoPropriedade.Pink,16000, 1200, 8000, 6000, 18000, 50000, 70000, 90000),
    new Propriedade(15,"Estação de Campanhã",TipoPropriedade.Estação, 20000, 2500, 10000, 0, 0, 0, 0, 0),
    new Propriedade(16,"Avenida da Boavista (Porto)",TipoPropriedade.Orange,18000, 1400, 9000, 7000, 20000, 55000, 75000, 95000),
    new CartasSorte(17,"Caixa da Comunidade"),
    new Propriedade(18,"Avenida da República (Lisboa)",TipoPropriedade.Orange, 18000, 1400, 9000, 7000, 20000, 55000, 75000, 95000),
    new Propriedade(19,"Rua de Sá da Bandeira (Porto)",TipoPropriedade.Orange, 20000, 1600, 10000, 8000, 22000, 60000, 80000, 100000),
    new EspacosEspeciais(20,"Estacionamento Livre"),
    new Propriedade(21,"Rua de Santa Catarina (Porto)",TipoPropriedade.Red, 22000, 1800, 11000, 9000, 25000, 70000, 87500, 105000),
    new CartasSorte(22,"SORTE"),
    new Propriedade(23,"Praça de Alvalade (Lisboa)",TipoPropriedade.Red, 22000, 1800, 11000, 9000, 25000, 70000, 87500, 105000),
    new Propriedade(24,"Rua Júlio Diniz (Porto)",TipoPropriedade.Red, 24000, 2000, 12000, 10000, 30000, 75000, 92500, 110000),
    new Propriedade(25,"Estação de S.Bento",TipoPropriedade.Estação, 20000, 2500, 10000, 0, 0, 0, 0, 0),
    new Propriedade(26,"Rua 31 de Janeiro (Porto)",TipoPropriedade.Yellow, 26000, 2200, 13000, 11000, 33000, 80000, 97500, 115000),
    new Propriedade(27,"Av. Fontes Pereira de Melo (Lisboa)",TipoPropriedade.Yellow, 26000, 2200, 13000, 11000, 33000, 80000, 97500, 115000),
    new Propriedade(28,"Companhia das Águas",TipoPropriedade.Companhia, 15000, 2000, 7500, 0, 0, 0, 0, 0),
    new Propriedade(29,"Avenida dos Aliados (Porto)",TipoPropriedade.Yellow, 28000, 2400, 14000, 12000, 36000, 85000, 102500, 120000),
    new EspacosEspeciais(30,"Vá para a Cadeia!"),
    new Propriedade(31,"Avenida da Liberdade (Lisboa)",TipoPropriedade.Green, 30000, 2600, 15000, 13000, 39000, 90000, 110000, 127500),
    new Propriedade(32,"Praça da Liberdade (Porto)",TipoPropriedade.Green, 30000, 2600, 15000, 13000, 39000, 90000, 110000, 127500),
    new CartasSorte(33,"Caixa da Comunidade"),
    new Propriedade(34,"Rua do Ouro (Lisboa)",TipoPropriedade.Green, 32000, 2800, 16000, 15000, 45000, 100000, 120000, 140000),
    new Propriedade(35,"Estação de St Apolónia",TipoPropriedade.Estação, 20000, 2500, 10000, 0, 0, 0, 0, 0),
    new CartasSorte(36,"SORTE"),
    new Propriedade(37,"Rua Augusta (Lisboa)",TipoPropriedade.Purple, 35000, 3500, 17500, 17500, 50000, 110000, 130000, 150000),
    new Imposto(38,"Imposto de Luxo",10000),
    new Propriedade(39,"Rossio (Lisboa)",TipoPropriedade.Purple, 40000, 5000, 20000, 20000, 60000, 140000, 170000, 200000),
};

```

Figura 6- Todos os Espaços do Tabuleiro

- Retira Cartas da Sorte

Nesta classe estão alocadas todas as cartas da sorte e a sua função, quanto retiram ou acrescentam ao jogador, status de prisão, etc.

```

1 referencia
public static class RetirarCartaSorte
{
    private static readonly Random rng = new Random();

    private static List<Func<Jogador, string>> listaCartasSorte = new List<Func<Jogador, string>>
    {
        PagueEmbriguez,
        RecueCasas,
        AvanceRossio,
        AvanceAvCentral,
        DespesasEscolares,
        VaiPreso,
        PalavrasCruzadas,
        Aniversario,
        AvanceCasaPartida,
        AvanceEstacaoCampanha,
        BancaDividendos,
        RecebaEmprestimoConstrucao,
        RecebaJurosAcoes,
        RecebaRendaAnual,
        PagueContaHospital,
        RecueCampoGrande,
        PagueSeguro,
        ReembolsoImpostoCapitais,
        ErroBanca,
        ContaMedico,
        Heranca,
        VendaStock,
        SegundoConcursoBeleza,
        MultaVelocidade,
        //ReparacoesRua,           NÃO IMPLEMENTADO
        //ReparacoesCasas,       NÃO IMPLEMENTADO
        AvanceJulioDiniz,
        //PagueMultaOuSorte,      NÃO IMPLEMENTADO
        //LivreCadeia,            NÃO IMPLEMENTADO
    };
}

```

Figura 7- Cartas da Sorte

- Cartas da Sorte

Esta classe faz um passa informações entre as cartas e um Form que gere as cartas e as escolhe de uma randômica.

```
namespace Monopoly.Classes
{
    7 referências
    public class CartasSorte : Espacos
    {
        6 referências
        public CartasSorte(int index, string nome) :base(index, nome)
        {
        }

        12 referências
        public override string ActOnPlayer(Jogador jogador)
        {
            return RetirarCartaSorte.GenerateRandomCard(jogador);
        }
    }
}
```

Figura 8- Retira uma Carta random do classe RetiraCartaSorte

- Propriedade

Nesta classe estão alocados todos os tipos de propriedades que cada jogador pode comprar, ou seja, estações, companhias e propriedades. Nesta classe também se trata da parte de rendas e hipotecas das mesmas.

```
43 referências
public class Propriedade : Espacos
{
    28 referências
    public Propriedade(int index, string nome, TipoPropriedade tipoPropriedade, int preco, int renda, int hipoteca, int renda1, int renda2, int renda3, int renda4, int renda5)
    {
        this.TipoPropriedade = tipoPropriedade;
        this.Preco = preco;
        this.Renda = renda;
        this.Dono = null;
        this.Hipoteca = hipoteca;
        this.Renda1 = renda1;
        this.Renda2 = renda2;
        this.Renda3 = renda3;
        this.Renda4 = renda4;
        this.Renda5 = renda5;
    }

    17 referências
    public TipoPropriedade TipoPropriedade { get; set; }
    18 referências
    public Jogador Dono { get; set; }
    3 referências
    public int Preco { get; set; }
    14 referências
    public int Renda { get; set; }
    5 referências
    public int Hipoteca { get; set; }
    7 referências
    public int Renda1 { get; set; }
    7 referências
    public int Renda2 { get; set; }
    7 referências
    public int Renda3 { get; set; }
    7 referências
    public int Renda4 { get; set; }
    7 referências
    public int Renda5 { get; set; }
    56 referências
    public int Casa / ref: set: 1 = 0;
```

Figura 9-Classe onde se gere toda a logística de propriedades

- Impostos

Esta classe trata de uma casa muito específica que cobra um certo valor quando o jogador passa por ela.

```
namespace Monopoly.Classes
{
    3 referências
    public class Imposto : Espacos
    {
        3 referências
        public int quantiaImposto { get; private set; }
        2 referências
        public Imposto(int index, string name, int quantiaImposto) :base(index,name)
        {
            this.quantiaImposto = quantiaImposto;
        }

        12 referências
        public override string ActOnPlayer(Jogador jogador)
        {
            Mensagem m = new Mensagem("Pague um imposto de " + this.quantiaImposto + " €!");
            m.ShowDialog();

            jogador.RetirarDinheiro(this.quantiaImposto);

            return this.Nome;
        }
    }
}
```

Figura 10- Retira dinheiro para pagar impostos quando o jogador passa na casa

- Jogador

Esta classe trata de tudo o que envolve o cada jogador, ou seja, o seu dinheiro, o seu posicionamento, o seu status de prisão, nome, peça, etc...

```

83 referências
public class Jogador
{
    public const int dinheiroInicial = 150000;
    public const int totalDeEspacos = 40;

    35 referências
    public Jogador(int index, string nome, Color cor, Image Peça)
    {
        this.Index = index;
        this.Nome = nome;
        this.cor = cor;
        this.Peca = Peça;
    }

    // Coloca o jogador numa nova posição
    10 referências
    public void SetPosicao(int novaPosicao)
    {
        int posicaoModificada = novaPosicao;

        // Se a nova posição for menor que zero
        if (posicaoModificada < 0)
        {
            // Soma 40 à posição
            posicaoModificada += totalDeEspacos;
        }
        // Se a nova posição for maior ou igual aos total de espacos (40)
        if (posicaoModificada >= totalDeEspacos)
        {
            // Retira 40 à nova posição e mostra a mensagem de passagem na casa Partida
            posicaoModificada = posicaoModificada - totalDeEspacos;
            Mensagem m = new Mensagem("Você passou pela casa partida e Recebeu 20.000€");
            m.ShowDialog();
            this.Depositar(20000);
        }
        // Coloca o jogador na nova posição
        this.posicaoAtual = posicaoModificada;
    }

    // Lista de propriedades de cada jogador
    99+ referências
    public List<Propriedade> Propriedades { get; set; } = new List<Propriedade>();

    // Guarda a posição atual do jogador, definindo a 0 no início do jogo.
    99+ referências
    public int posicaoAtual { get; set; } = 0;

    // -----
    10 referências
    public int Index { get; private set; }

    // Boolean para saber se está na cadeia, definido como false no início
    18 referências
    public bool estaPreso { get; set; } = false;

    // Dinheiro do Jogador, definindo como 150,000 no início
    11 referências
    public int Dinheiro { get; private set; } = dinheiroInicial;

    9 referências
    public int turnosPrisao { get; set; } = 0;

    3 referências
    public int doubleCount { get; set; } = 0;

    99+ referências
    public Color cor { get; private set; }

    10 referências
    public string Nome { get; private set; }
    99+ referências
    public Image Peca { get; private set; }

    3 referências
    public bool falencia { get; set; } = false;
}

```

Figura 11- Formatação do Jogador

- Espaços Especiais

Estes espaços são basicamente as esquinas do tabuleiro onde se passam basicamente 3 coisas:

- 1º. Estar na Partida;
- 2º. Estar na Prisão provisoriamente ou por 3 jogadas, ou menos;
- 3º. Estar no Parque de estacionamento onde não acontece nada;

```
namespace Monopoly.Classes
{
    5 referências
    public class EspacosEspeciais : Espacos
    {
        4 referências
        public EspacosEspeciais(int index, string nome) : base(index, nome)
        {
        }
        12 referências
        public override string ActOnPlayer(Jogador jogador)
        {
            if (this.Index == 0)
            {
                return "Estás na Casa Partida!";
            }
            else if (this.Index == 10)
            {
                if (jogador.estaPreso == false)
                {
                    return "Estás a visitar a Cadeia";
                }
                else
                {
                    return "Estás na Cadeia";
                }
            }
            else if (this.Index == 20)
            {
                return "Estás no Parque de Estacionamento. Não acontece nada.";
            }
            else
            {
                jogador.estaPreso = true;
                jogador.turnosPriso = 3;
                jogador.SetPosicao(10);
                return "Estás na Prisão! Não vais poder jogar nos proximos 3 turnos.";
            }
        }
    }
}
```

Figura 12- Espaços especiais são resumidamente todas as esquinas do tabuleiro

- Tipo de Propriedade

```
33 referências
public enum TipoPropriedade
{
    Brown, Blue, Pink, Orange, Red, Yellow, Green, Purple, Estação, Companhia
}
```

Figura 13- Guarda as cores associadas ao jogador do tipo de propriedades (Estações, Companhias ou Propriedades)

3.1 Fluxograma

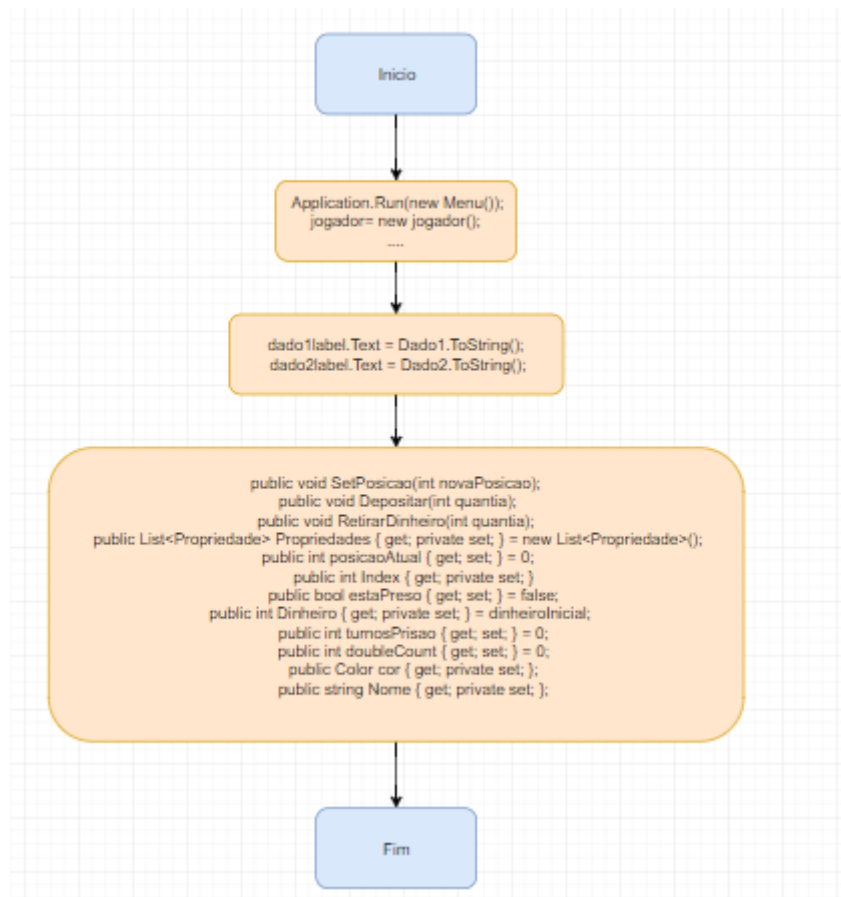


Figura 14- Fluxograma que Representa o funcionamento do Programa

4. Finalização do Jogo

4.1 Dificuldades na realização

Neste momento podemos dizer que o jogo está a funcionar a 95% do seu potencial, claro que deveria conter mais algumas melhorias mas para projeto final sentimo-nos realizados. Na execução do mesmo os erros/BUGS foram muitos desde a os fluxos de dinheiros, o movimento dos jogadores nas casas, o aparecimento de casas e hotéis no tabuleiro, a execução de várias cartas da sorte e o *multiplayer* DE 8 jogadores.

Mas todos esses retrocessos conseguiram ser ultrapassados com muitas horas de estudo e prática para tentar por a executar o programa.

4.2 Interface



Figura 15- Menu inicial do Jogo



Figura 16-Menu (continuação, botão Sobre) Curiosidades do jogo

Nesta parte todos os jogadores devem escolher nomes, peças e cores diferentes ou o programa irá emitir um erro e só executará quando todos tiverem nomes, peças e cores diferentes.

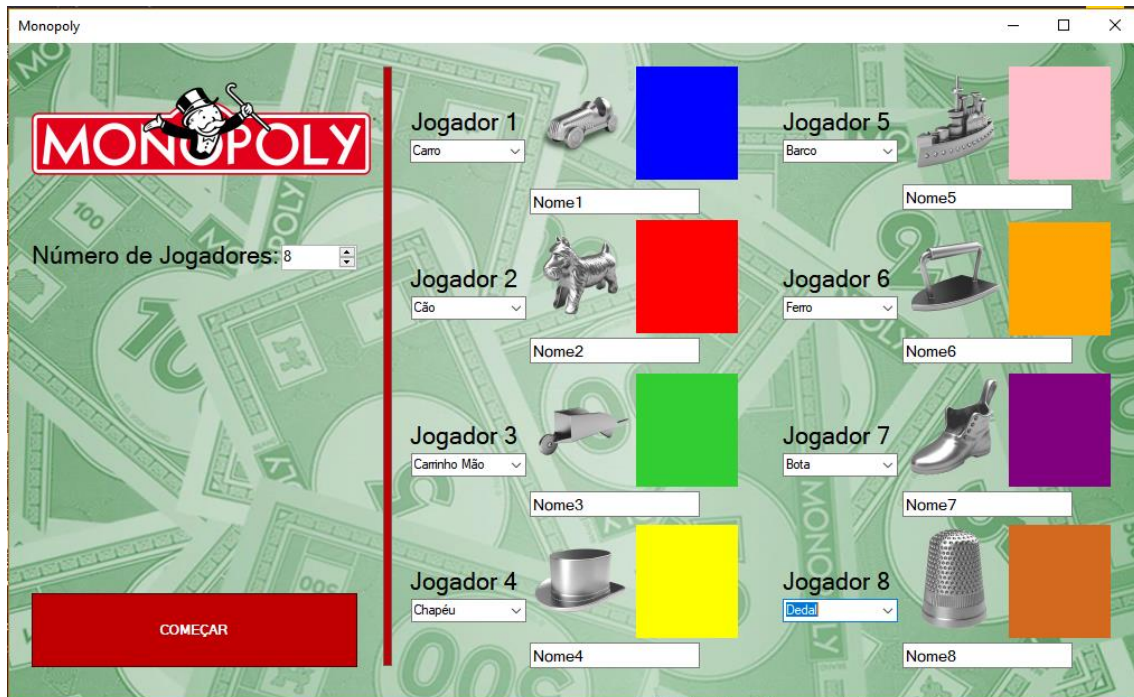


Figura 17- Escolha MultiPlayer até 8 jogadores máximos

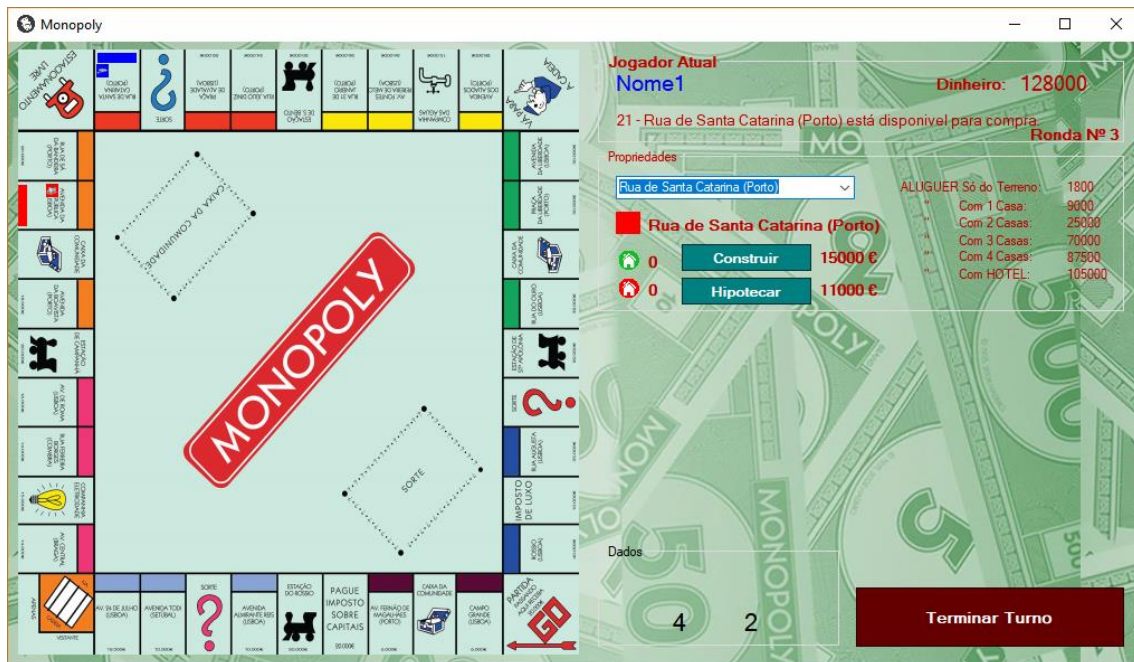


Figura 18- Modo Jogo



Figura 19- a Sequencia destas duas imagens são uma janela que abre quando o jogador calha numa casa da Sorte.



Figura 20- Message Box caso o jogador tenha as 3 casas para poder construir Casas e Hotéis

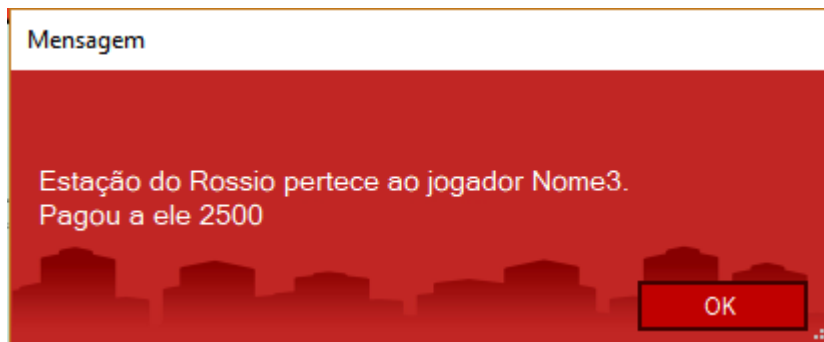


Figura 21-Messagebox que dá ao jogador mensagens variadas



Figura 22- Fim onde reconhece o vencedor e dá algumas opções extras

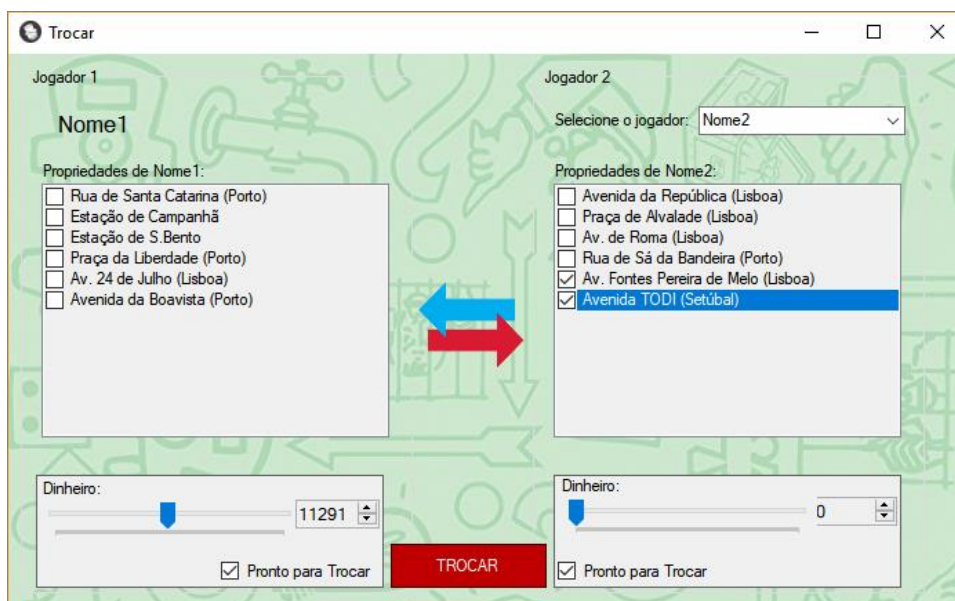


Figura 8- Troca de Cartas entre jogadores

5. Conclusão

Para concluir, podemos afirmar que os objetivos que foram expostos na *Descrição do Jogo* foram concluídos com sucesso. No final destas semanas de Trabalho sentimo-nos realizados após termos conseguido cumprir este objetivo. Por vezes achamos que se iria tornar impossível pois as dificuldades neste projeto foram imensas e o tempo contrarrelógio e os vários trabalhos não ajudaram na rápida realização do mesmo. Queremos também nesta conclusão agradecer ao Professor Romero que nos concebeu um prazo alargado para conseguir melhorar o nosso projeto promissor.

Em conclusão foi um trabalho prático que apresentou um enorme desafio e que nos ajudou a melhorar as nossas técnicas de programador e de por à prova as matérias que aprendemos durante o semestre e a nossa capacidade de adaptamento e resolução de problemas.