

Projeto Semana de Informática

Jogo Multiplayer

Curso de Jogos Digitais – Módulo 4
2º semestre de 2015

Objetivo do Projeto

Construção de um jogo multiplayer, totalmente desenvolvido em C#, sem utilização de frameworks e biblioteca de jogos de terceiros. Nesse projeto deverão ser construídos dois programas:

1. **Cliente:** programa que permitirá que um usuário (Jogador 1) jogue contra outro (Jogador 2). Esse programa funcionará em dois modos distintos:
 - 1.1. **P2P** (*Peer-to-peer* ou ponto-a-ponto): permite a conexão direta entre dois computadores nos quais os usuários irão competir entre si;
 - 1.2. **Campeonato:** embora o jogo se desenvolva sempre em uma conexão ponto-a-ponto (P2P), nesse modo, os jogadores serão selecionados através de um servidor que irá controlar o campeonato de acordo com regras definidas nesse documento.
2. **Servidor:** programa para gerenciar os campeonatos entre diversos jogadores.

Componentes Curriculares

- Oficina de Desenvolvimento de Jogos Multiplayer (Prof Francisco Calaça)
- Programação Concorrente (Prof. Willians Santos)
- Redes de Computadores (Prof. Dinailton José)

O Jogo

O jogo é uma adaptação da clássica brincadeira de “rouba-bandeira”, em tempo real, onde dois jogadores remotos se enfrentam para tentar capturar a bandeira do jogador adversário e retorná-la à sua zona inicial, antes que o jogador adversário o faça ou antes que o tempo se acabe. Ao mover o seu jogador, esse movimento é replicado na tela do dispositivo do seu adversário, de forma que os jogadores estarão sempre vendo, remotamente, a mesma tela. Cada jogador poderá atirar nas direções horizontal e vertical, indefinidamente, visando destruir obstáculos à sua frente (um único tiro), ou inativar o jogador adversário (três tiros). Um jogador permanece inativo por um determinado intervalo de tempo, onde nesse período de tempo dele não poderá movimentar-se, nem atirar.

A Figura 1 mostra as dimensões do campo de batalha, em termos do número de pixels existentes na tela do jogador (LARG x ALT). O jogo deverá ser implementado para funcionar em dispositivos com telas de tamanho e resoluções distintos. A Figura 2 mostra a configuração inicial do campo

de batalha. O elemento “célula de energia” aumenta a velocidade de movimento do jogador que o pegar primeiro em 50% até que a partida termine ou até que o jogador que o pegou seja inativado (levar 3 tiros). Nesse caso, uma nova célula surgirá na sua posição inicial. Caso nenhum jogador consiga retornar à sua zona inicial em até 2 minutos, será considerado vencedor o jogador que esteja de posse da bandeira do adversário. Caso os dois jogadores já tenham pego as bandeiras, será vencedor o jogador que tiver de posse da célula de energia. Caso contrário, o jogo terá um empate. O jogo funciona em um sistema de melhor de três. Ao iniciar cada partida, o jogo volta à sua configuração inicial.



Figura 1- Dimensões do campo de batalha

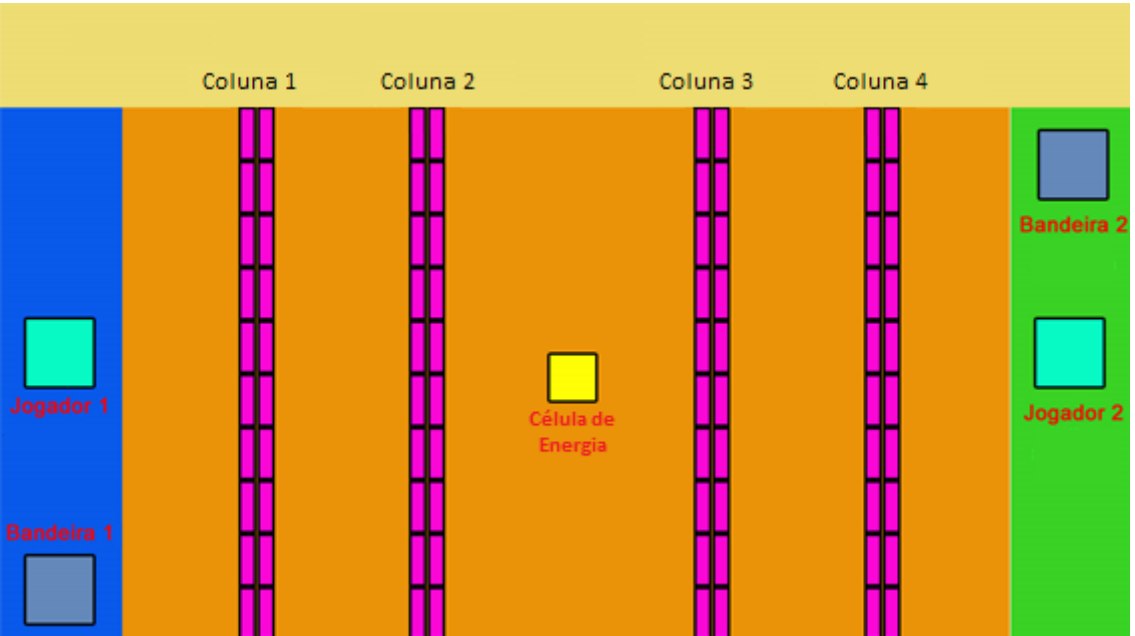


Figura 2- Configuração inicial do campo de batalha

Na parte superior da tela (HUD – *heads-up display*) serão mostrados dados referentes ao jogo, sendo que deverá ser mostrado, pelo menos:

- Identificação de cada jogador
- Tempo restante para o término da partida
- Placar e vidas restantes ao jogador
- Caso o jogador esteja inativo, tempo restante para ele voltar à ativa

As cores e formatos dos elementos gráficos da tela, serão livres, ou seja, cada implementação do jogo poderá fazer as suas próprias escolhas, desde que respeitem as dimensões definidas. Também serão livres os efeitos do jogo (imagens e sons).

Posicionamento Inicial e Dimensões dos Elementos do Jogo

- **Jogadores:** o “Jogador 1” será o jogador que convidou para o jogo e o “Jogador 2” será o jogador que aceitou o convite.
 - **Posição inicial:** no centro da sua respectiva área. A posição de cada jogador será o centro do quadrado envolvente do jogador.
 - **Dimensões:** 0.05 ALT x 0.05 ALT (quadrado com 5% da altura da tela)
- **Bandeiras**
 - **Posições iniciais:**
 - Bandeira 1: horizontalmente centralizado em relação à área do jogador e a parte inferior a 0.02 ALT (2% da altura da tela) da parte inferior da área do jogador.
 - Bandeira 2: horizontalmente centralizado em relação à área do jogador e a parte superior a 0.02 ALT (2% da altura da tela) da parte superior da área do jogador.
 - **Dimensões:** 0.05 ALT x 0.05 ALT (mesmo tamanho dos jogadores)
- **Célula de Energia Extra**
 - **Posição inicial:** no centro da zona de combate
 - **Dimensões:** 0.03 ALT x 0.03 ALT (quadrado com 3% da altura da tela)
- **Colunas de obstáculos:** existirão 4 colunas compostas de duas fileiras de blocos com 10 blocos em cada fileira, tendo cada bloco o mesmo tamanho. Não existe espaço entre as fileiras, nem entre os blocos de uma mesma fileira.
 - **Posições iniciais:** considerando o lado esquerdo da primeira fileira de cada coluna.
 - Coluna 1: a 15% do lado esquerdo da zona de combate
 - Coluna 2: a 30% do lado esquerdo da zona de combate
 - Coluna 3: a 70% do lado esquerdo da zona de combate
 - Coluna 4: a 85% do lado esquerdo da zona de combate

- **Dimensões:** cada bloco será um retângulo com altura de 0.1 Altura da Zona de Combate e largura de 0.01 LARG (largura da tela)
- **Tiro:** quadrado com 50% do tamanho do jogador

IMPORTANTE: como o jogo deverá ser executado em qualquer dispositivo, independentemente da sua resolução e dimensão (quantidade de pixels), internamente, o jogo deverá trabalhar com coordenadas normalizadas.

Dinâmica do Jogo

- A colisão dos elementos do jogo será determinada pelo retângulo envolvente de cada elemento, respeitadas as dimensões especificadas.
- O jogador deverá detectar colisão com as bandeiras, obstáculos, célula de energia, com o outro jogador e com os limites da área do jogo.
- Existirá somente uma célula de energia extra no jogo, ou seja, quem pegar ele, fica com ele até que ganhe o jogo ou seja inativado. Nesse caso, a célula de energia voltará para a sua posição inicial.
- Os jogadores não podem atirar dentro das áreas dos jogadores (área própria ou área do adversário).
- A velocidade normal do jogador é de $(0.10 \text{ LARG})/\text{segundo}$ (10% da largura da tela); de posse da célula de energia, sua velocidade será de $(0.15 \text{ LARG})/\text{s}$; a velocidade do tiro é de $(0.25 \text{ LARG})/\text{s}$.
- O tiro possui detecção de colisão com os obstáculos, jogadores, célula de energia e com os limites da tela/interface.
- Caso o tiro acerte a célula de energia, o tiro desaparece e nada acontece.
- O jogador pode atirar a cada 0.5 segundos e a quantidade de balas é infinita.
- O jogador não sofrerá danos ao tocar no adversário, nem nos obstáculos.
- Para ganhar o jogo, o jogador que carrega a bandeira deve estar completamente dentro de sua área inicial.
- O período no qual um jogador permanece inativo é igual à 5 segundos multiplicado pela quantidade de vezes que este já ficou inativo.
- Ao ser reativado, o jogador fica invulnerável por 3 segundos.

O Campeonato

Os jogadores poderão participar de um campeonato. Para isso, um servidor deverá ser usado para gerenciar o campeonato. Eis as regras do campeonato:

1. O campeonato ocorrerá em uma data e horário específicos, de forma online;
2. Os jogadores deverão se inscreverem previamente no campeonato;
3. Na hora marcada para o início do campeonato, o servidor irá verificar quais jogadores estão online e irá definir quem enfrentará quem;

4. As disputas ocorrerão nos mesmos moldes das disputas ponto-a-ponto, conforme descrita anteriormente (melhor de 3 batalhas, com ou sem tempo limite) e definido nas configurações do servidor;
5. Os perdedores de cada disputa estão automaticamente eliminados do campeonato;
6. O servidor irá selecionar, dentre os vencedores, a próxima rodada do campeonato;
7. Se em uma rodada do campeonato existir um número ímpar de jogadores, o servidor deverá sortear um jogador que estará automaticamente classificado para a próxima rodada;
8. O processo repete até que haja somente um vencedor.

Protocolo de Comunicação

As mensagens trocadas pelos dispositivos serão sempre no formato ASCII, onde cada mensagem possui um cabeçalho de tamanho fixo e uma parte variável, que irá depender do tipo da mensagem trocada entre os dispositivos. O cabeçalho de cada mensagem possui tamanho fixo (5 caracteres) e é formado pelos seguintes campos:

- ID: 2 dígitos que identifica o tipo da mensagem
- Tamanho: 3 dígitos que indica o número de caracteres (tamanho) total da mensagem, incluindo os 5 dígitos do cabeçalho.

Mensagens trocadas antes do Jogo

Mensagens **UDP** trocadas entre o programa corrente e demais dispositivos com o objetivo de saber quem está online e fazer o convite para participar de um jogo. O programa irá utilizar a porta 20152 (uma referência para 2015/2) para escutar estas mensagens. Essa porta deverá ficar disponível para troca de mensagens, mesmo quando o jogador estiver participando de um jogo. O tipo e formato das mensagens estão descritos a seguir:

- Mensagem **"01"** (MSG01): saber quem está online.
Formato da Mensagem: **01**999Apelido|Nome-completo
01: identificação da mensagem
999: tamanho total da mensagem
Apelido: no máximo 15 caracteres contendo o nome no qual você deseja ser identificado.
|: separador de campo
Nome-do-jogador: seu nome completo (pode ser utilizado pelo programa remoto para de identificação de dois jogadores com apelidos iguais ou estranhos).

Obs.: essa mensagem será enviada via broadcast para a porta 20152 de todos os dispositivos da rede sempre que desejar saber quais são os jogadores que estão online no momento, sendo que o programa poderá enviar essa mensagem periodicamente enquanto estiver em execução. Cada programa que receber essa mensagem irá responder com uma MSG02 se identificando, permitindo que seja montada uma lista de

jogadores que estão online. Essa lista deverá ser usada para convidar jogadores para a disputa de um jogo.

- Mensagem “02” (MSG02): resposta à MSG01 informando que você está online

Formato da Mensagem: **02**999Apelido|Nome-completo

02: identificação da mensagem

999: tamanho total da mensagem

Apelido: no máximo 15 caracteres contendo a identificação do jogador

|: separador de campo

Nome-do-jogador: nome completo do jogador

Obs.: cada jogador que receber a mensagem MSG01 deverá responder com essa mensagem somente para quem enviou a mensagem original, se identificando, mostrando que você está online. Essa mensagem também poderá ser enviada periodicamente para informar que você continua online, mesmo que não haja uma MSG01 original.

- Mensagem “03” (MSG03): convite para participar de um jogo

Formato da Mensagem: **03**999Apelido

03: identificação da mensagem

999: tamanho total da mensagem

Apelido: no máximo 15 caracteres contendo a identificação do jogador.

Obs.: essa mensagem será enviada para um jogador específico que já tenha respondido uma MSG01. O recebimento dessa mensagem indica que o jogador (Apelido) está te convidando para um jogo.

- Mensagem “04” (MSG04): resposta à MSG03 indicando se convite foi aceito ou não

Formato da Mensagem: **04**999Apelido|Porta

04: identificação da mensagem

999: tamanho total da mensagem

Apelido: no máximo 15 caracteres contendo a identificação do jogador

|: separador de campo

Porta: caso o convite foi aceito, número da porta TCP que será utilizada para fazer a conexão P2P. Caso não queira (ou não possa) aceitar o convite, o número da porta deverá ser 0 (zero) indicando que o você não quer jogar ou que já está jogando com outro jogador.

Obs.: essa mensagem será enviada somente para quem te enviou a MSG03. Ao aceitar um convite, o programa deverá abrir um socket TCP que ficará escutando a porta informada na mensagem, esperando que o jogador que fez o convite conecte com você para iniciar o jogo.

Mensagens trocadas durante o Jogo

Mensagens **TCP** trocadas entre dois jogadores que aceitaram participar de um jogo. Essas mensagens serão enviadas quando houver necessidade de alteração de elementos gráficos na

tela (exceto movimentação de tiro). Para manter o sincronismo entre as telas dos dois jogadores, antes de fazer a atualização da tela local, a mensagem correspondente deverá ser enviada para o jogador remoto e, somente após receber a confirmação de recebimento da mensagem, é que a tela local será atualizada. A tela do jogador remoto será atualizada assim que essa mensagem for recebida.

O tipo e formato das mensagens estão descritos a seguir:

- Mensagem “**10**” (MSG10): início de jogo

Formato da Mensagem: **10005**

10: identificação da mensagem

005: tamanho total da mensagem

Obs.: indica que jogador está pronto para iniciar o jogo. Jogo somente poderá ser iniciado quando o outro jogador também enviar MSG10, ou seja, os dois jogadores deverão enviar e receber uma mensagem MSG10.

- Mensagem “**11**” (MSG11): informa que houve uma movimentação do jogador.

Formato da Mensagem: **11999X|Y|Direção**

11: identificação da mensagem

999: tamanho total da mensagem

X: nova posição horizontal do jogador (em coordenadas normalizadas)

|: separador de campo

Y: nova posição vertical do jogador (em coordenadas normalizadas)

Direção: direção na qual jogador “caminha” (ou “aponta”), podendo ser:

- **B**: para baixo
- **C**: para cima
- **D**: para a direita
- **E**: para a esquerda

Obs.: a utilização de coordenadas normalizadas (valores das coordenadas entre 0..1) permite a utilização de dispositivos com resoluções e dimensões diferentes. Essa mensagem deverá ser enviada ANTES do programa local atualizar a posição do jogador na tela. Essa atualização somente poderá ser feita depois que a mensagem de confirmação MSG12 for recebida.

- Mensagem “**12**” (MSG12): resposta à mensagem MSG11, confirmando a posição do jogador remoto.

Formato da Mensagem: **12999X1|Y1|Direção1|X2|Y2|Direção2**

12: identificação da mensagem

999: tamanho total da mensagem

X1|Y1|Direção1: parâmetros originais da mensagem MSG11, confirmando que o posicionamento do jogador remoto foi confirmado.

X2|Y2|Direção2: parâmetros opcionais indicando o novo posicionamento do jogador remoto.

Obs.: os parâmetros **X2|Y2|Direção2** poderão ser enviados (opcionalmente) na mensagem de resposta (se o posicionamento do jogador tiver sido alterado desde a última MSG11 enviada) ou em uma nova mensagem MSG11. Ao receber uma

mensagem MSG12 com esses parâmetros, o programa deverá responder com uma nova mensagem MSG12 confirmando o posicionamento do jogador remoto para que ele possa atualizar sua tela.

- Mensagem “13” (MSG13): foi disparado um tiro

Formato da Mensagem: **13**999X|Y|Direção|IDTiro

13: identificação da mensagem

999: tamanho total da mensagem

X: posição horizontal da qual o tiro foi disparado (em coordenadas normalizadas)

|: separador de campo

Y: posição vertical da qual o tiro foi disparado (em coordenadas normalizadas)

Direção: direção na qual o tiro “caminha”, podendo ser:

- **B**: para baixo
- **C**: para cima
- **D**: para a direita
- **E**: para a esquerda

IDTiro: número sequencial, iniciado em 1, que identifica o tiro que foi disparado.

A cada tiro disparado, esse número deverá ser incrementado, ou seja, ele é um contador de tiros disparados.

Obs.: ao receber a notificação de tiro, o programa remoto irá controlar localmente o deslocamento do tiro, com velocidade constante de $(0.25 * \text{largura da tela})$ por segundo. Até que o tiro colida com algum objeto da tela, não será enviada nenhuma outra mensagem referente a esse tiro. Somente o programa que disparou o tiro é que irá enviar mensagem de colisão desse tiro com algum objeto.

- Mensagem “14” (MSG14): resposta à mensagem MSG13, confirmando o disparo do jogador remoto.

Formato da Mensagem: **14**999X1|Y1|Direção1|IDTiro1|X2|Y2|Direção2|IDTiro2

14: identificação da mensagem

999: tamanho total da mensagem

X1|Y1|Direção1|IDTiro1: parâmetros originais da mensagem MSG13, confirmando que o tiro foi disparado.

X2|Y2|Direção2|IDTiro2: parâmetros opcionais indicando disparo do jogador remoto.

Obs.: os parâmetros **X2|Y2|Direção2|IDTiro2** poderão ser enviados (opcionalmente) na mensagem de resposta (se o jogador tiver atirado desde a última MSG13 enviada) ou em uma nova mensagem MSG13. Ao receber uma mensagem MSG14 com esses parâmetros, o programa deverá responder com uma nova mensagem MSG14 confirmando o disparo do jogador remoto para que ele possa atualizar sua tela.

- Mensagem “15” (MSG15): colisão entre objetos do jogo

Formato da Mensagem: **15**999Obj1|Obj2|IDTiro

01: identificação da mensagem

999: tamanho total da mensagem

Obj1: identificação do objeto que provocou a colisão. Cada objeto da tela que terá detecção de colisão tem identificação própria, a saber:

- **J1:** jogador 1
- **J2:** jogador 2
- **B1:** bandeira 1
- **B2:** bandeira 2
- **T:** tiro
- **CE:** célula de energia
- **BLCFNN:** um bloco constituinte das colunas de obstáculos, onde 'BL' indica colisão com um bloco e "CFNN" identifica qual bloco sofreu colisão, a saber:
 - **C:** qual coluna (1 a 4)
 - **F:** qual fileira da coluna (1 ou 2)
 - **NN:** qual bloco da fileira, enumerado de 01 a 10, de cima para baixo (cada fileira possui 10 blocos).

|: separador de campo

Obj2: identificação do objeto que foi colido pelo primeiro objeto (Obj1). Segue o mesmo padrão que foi utilizado para identificação própria do Obj1.

ID: parâmetro opcional, devendo ser enviado somente no caso do Obj1 ser um tiro, identificando qual foi o tiro que colidiu com o Obj2

Obs.: mensagens de colisões entre um tiro e um objeto somente serão enviadas pelo programa que fez o disparo, ou seja, programa que enviou uma mensagem MSG13. Se o programa identificar a colisão entre um tiro que não foi ele que disparou e algum objeto, ele deverá ignorar a colisão e esperar que uma mensagem de colisão desse tiro chegue para que a tela seja atualizada. Exemplos de mensagens de colisão:

- Jogador 1 pega a bandeira: **15010J1|B2**
- Tiro número 2 acerta o penúltimo bloco da primeira fileira da terceira coluna: **15013T|BL3109|2**
- Jogador 2 pega a célula de energia: **15010J2|CE**

- Mensagem "**16**" (MSG16): confirmação de colisão entre objetos do jogo

Formato da Mensagem: **16999Obj1|Obj2|ID**

16: identificação da mensagem

999: tamanho total da mensagem

Obj1|Obj2: parâmetros originais da mensagem MSG15, confirmando a colisão entre os objetos Obj1 e Obj2.

ID: parâmetro opcional, devendo ser enviado somente no caso do Obj1 ser um tiro, identificando qual foi o tiro que colidiu com o Obj2

Obs.: somente DEPOIS de receber a confirmação da colisão é que o programa poderá atualizar a tela conforme os objetos envolvidos na colisão.

- Mensagem "**17**" (MSG17): ativação/inativação de jogador

Formato da Mensagem: **17006N**

17: identificação da mensagem

006: tamanho total da mensagem

N: indica se jogador foi congelado/inativado (0) ou foi reativado (1)

Obs.: o jogador será inativado (congelado por um tempo determinado) quando ele receber o terceiro tiro. Como os tiros são controlados pelo programa que fez o disparo, a mensagem de inativação do jogador (status 0) será enviada por esse programa. Ao receber essa mensagem, o programa que teve o jogador congelado irá esperar esse período de tempo de congelamento e, quando o jogador for reativado/descongelado, ele deverá enviar uma mensagem de ativação do seu jogador (status 1) para o programa remoto.

- Mensagem “**18**” (MSG18): confirmação da ativação/inativação de jogador

Formato da Mensagem: **18006N**

18: identificação da mensagem

006: tamanho total da mensagem

N: indica se jogador foi congelado/inativado (0) ou foi reativado (1)

Obs.: somente depois de receber essa mensagem de confirmação é que o programa que enviou a mensagem MSG17 irá congelar ou descongelar o jogador.

- Mensagem “**19**” (MSG19): fim de jogo

Formato da Mensagem: **19005**

19: identificação da mensagem

005: tamanho total da mensagem

Obs.: indica que o jogo terminou. Isso pode ocorrer em duas situações: quando um jogador entrou na sua área carregando a bandeira, ou quando o jogador desiste do jogo. Não haverá necessidade de confirmação dessa mensagem.

Mensagens trocadas entre o Servidor e os Clientes (Campeonatos)

[MENSAGENS AINDA NÃO FORAM DEFINIDAS. AGUARDE ATUALIZAÇÃO DESSE DOCUMENTO]