

PROJETO FINAL

1. Objetivo

O objetivo deste trabalho é a implementação de um jogo eletrônico usando elementos gráficos em duas dimensões. Nele os alunos vão desenvolver suas competências no uso de estruturas de repetição, vetores, entrada e saída de dados e programação estruturada como um todo.

2. A equipe de desenvolvimento

O trabalho será desenvolvido por uma equipe de 2 ou 3 alunos, em que 1 deles será o líder. Para formar uma equipe, um dos integrantes deve publicar no fórum “Formação de grupos para projeto”, no SIGAA, uma mensagem contendo o nome e a sub-turma de cada integrante, bem como especificando quem é o líder da equipe.

As equipes poderão ser criadas ou modificadas até o dia 03/05. Para isso basta que o aluno substitua a publicação no fórum (apagando a antiga e criando uma nova). Se houverem publicações conflitantes (o mesmo aluno em duas equipes), valerá a mais antiga. Após o dia 03/05, as equipes não poderão sofrer alterações. Qualquer aluno sem equipe; ou em uma equipe com menos do que 2 e mais do que 3 alunos; ou em equipe sem líder; receberá nota 0 no projeto.

As equipes podem ser formadas por alunos de diferentes sub-turmas, mas a apresentação do projeto ocorrerá no horário da aula de laboratório do líder da equipe. A não participação por conflito de horário é de responsabilidade do aluno.

3. O projeto de programação

O jogo eletrônico será desenvolvido em 8 etapas. A ferramenta de desenvolvimento proposta é a ferramenta Processing (<https://processing.org/>). O jogo será construído com elementos gráficos em duas dimensões e o com tema de livre escolha, mas que atenda as especificações a seguir:

1ª etapa (10%): Ao iniciar o programa, deve aparecer na tela uma figura geométrica ou uma imagem. Essa figura será o Jogador e deve ser posicionada no meio de uma das bordas da janela.

2ª etapa (10%): Movimentar o Jogador utilizando o teclado. Nenhuma parte do Jogador deve sair da tela.

3ª etapa (15%): Em um tempo aleatório, deve aparecer na tela uma figura geométrica ou uma imagem que será um Obstáculo posicionado em algum ponto aleatório da borda oposta à posição inicial do Jogador. Por exemplo, se o Jogador inicialmente está na borda esquerda, o Obstáculo deve estar na borda direita.

4ª etapa (15%): O Obstáculo deve atravessar a tela indo da sua posição inicial até a borda oposta. Ao sumir da tela, deve reaparecer na sua borda original em um ponto aleatório e em tempo aleatório.

5ª etapa (15%): Adicionar um número arbitrário de Obstáculos com o mesmo comportamento descrito nas 3ª e 4ª tarefas. Nessa etapa é obrigatório o uso de vetores e estruturas de repetição.

6ª etapa (15%): Parar todos os objetos da tela quando o Jogador colidir com qualquer um dos Obstáculos e diminuir as vidas do jogador .

7ª etapa (10%): Trocar entre telas do jogo. O jogo deve possuir no mínimo três momentos: tela de inicialização contendo o nome do jogo, por exemplo; tela de andamento do jogo, onde o usuário pode interagir com o personagem do jogo; e tela de fim de jogo (*Game Over*).

8ª etapa (10%): Apresentar uma estratégia de pontuação e aumentar o nível de dificuldade do jogo de acordo com faixas de pontos atingidas. Assim, a media que o jogador for avançando no jogo a dificuldade vai aumentando também.

Pontuação extra (5% cada):

- Adicionar uma animação quando houver uma colisão
- Permitir que o usuário com algum comando reinicie o jogo após uma colisão
- Adicionar na tela informações sobre o jogo, tais como pontuação, tempo, nível de dificuldade, vidas, etc
- Adicionar um mecanismo de disparo para que o Jogador possa destruir os Obstáculos

Observação: A implementação do projeto deve ser gradativa e a medida que novas etapas forem concluídas as anteriores devem ser mantidas. Ao final da implementação, depois de concluídas todas as etapas, o jogo deverá funcionar com todos os requisitos especificados em cada etapa.

4. Cronograma

Data	Horário	Local	Descrição
03-05			Fim do período para formação ou alteração de equipes.
13-05			Entrega, via SIGAA, da 1ª, 2ª e 3ª Etapa
24-05	3T56	Aud. C	Acompanhamento
25-05	4M12	Lab. 2	Acompanhamento

26-05	-	-	Feriado
27-05	6M12	Lab. 3	Acompanhamento
31-05	3T56	Aud. C	Acompanhamento
01-06	4M12	Lab. 2	Acompanhamento
02-06	5T56	Aud. C	Acompanhamento
03-06	6M12	Lab. 3	Acompanhamento e entrega, via SIGAA, da 4 ^a , 5 ^a e 6 ^a Etapa
07-06	3T56	Aud. C	Acompanhamento
08-06	4M12	Lab. 2	Acompanhamento
09-06	5T56	Aud. C	Acompanhamento
10-06	6M12	Lab	Acompanhamento e entrega, via SIGAA, do código-fonte da versão final do projeto.
11-06			Entrega, via SIGAA, do relatório final do projeto.
14-06	3T56	Aud. C	Apresentações do projeto
15-06	4M12	Lab. 2	Apresentações do projeto
16-06	5T56	Aud. C	Apresentações do projeto
17-06	6M12	Lab. 3	Apresentações do projeto

5. Plantões de dúvidas

Durante o período de desenvolvimento do projeto algumas aulas teóricas/práticas serão substituídas por plantões de dúvidas. Haverão plantões de dúvidas em laboratório nos dias 25/05, 27/05, 01/06, 03/06, 08/06 e 10/06 das 07h as 8:40h nos laboratórios 2 (quartas e sextas).

A ordem de atendimento será definida através de sorteio para as equipes presentes no início da aula. As equipes que chegarem após o início da aula entram no final da fila.

Também haverão plantões de dúvidas no auditório nos horários reservado para aula. Esse cronograma seguirá de acordo com o disponibilizado no SIGAA.

6. Entrega

A entrega dos códigos-fonte e do relatório ocorrerão até as 23h59 das respectivas datas definidas no cronograma desse documento (Seção 4).

As tarefas, no SIGAA, serão configuradas para permitirem reenvio. Ou seja, a

equipe pode enviar o código-fonte e/ou relatório várias vezes, fazendo valer sempre o último envio.

A perda do prazo de envio de qualquer tarefa por qualquer motivo será interpretada como **desistência** da equipe no projeto.

7. Presença e avaliação

Os plantões de dúvidas e a apresentação do projeto contarão como aulas na carga horária da disciplina e haverá lista de presença. Será considerado presente em todos os plantões de dúvidas o aluno que entregar todas as etapas e apresentar o projeto.

Cada projeto será avaliado através da execução do jogo, da inspeção do código-fonte (organização e clareza da lógica utilizada), da apresentação oral e do relatório.

No momento da apresentação, todos os integrantes devem entender a lógica e funcionamento do jogo, e não somente a parte que desenvolveram. A avaliação do relatório será feita como parte da apresentação do líder.

O relatório deve ser elaborado pelo líder da equipe seguindo o modelo do arquivo “Relatório – modelo.pdf”, que será publicado no SIGAA.

O projeto tem uma pontuação máxima de 6.0 pontos na unidade 2, divididos da seguinte maneira:

1. Pontuação para cada etapa:
 1. Etapa 1 (E1) vale 10% da nota: 0.6;
 2. Etapa 2 (E2) vale 10% da nota: 0.6;
 3. Etapa 3 (E3) vale 15% da nota: 0.9;
 4. Etapa 4 (E4) vale 15% da nota: 0.9;
 5. Etapa 5 (E5) vale 15% da nota: 0.9;
 6. Etapa 6 (E6) vale 15% da nota: 0.9;
 7. Etapa 7 (E7) vale 10% da nota: 0.6;
 8. Etapa 8 (E8) vale 10% da nota: 0.6;
2. A soma $E1 + E2 + E3 + E4 + E5 + E6 + E7 + E8$ é a pontuação base da equipe.
3. A apresentação (AP) tem nota de 0 a 10 atribuída para cada aluno individualmente sendo utilizada para ponderar a nota final do aluno.
4. A nota do projeto (NP) é calculada pela expressão: $NP = ((E1 + E2 + E3 + E4 + E5 + E6 + E7 + E8) * AP) / 10$.

Erros de compilação, erros de execução e um código-fonte de baixa qualidade poderão penalizar a equipe como um todo. Faltar no dia da apresentação, colaborar pouco para com o projeto ou não apresentar satisfatoriamente bem o trabalho poderão penalizar o integrante individualmente. O relatório incompleto ou incoerente pode penalizar o líder da equipe.

8. Caracterização do Jogo

Trata-se de um jogo no qual o objetivo é manter o personagem principal maior tempo possível sem colidir com qualquer um dos obstáculos. A escolha do personagem principal e dos obstáculos fica a critério do desenvolvedor do jogo. Algumas possibilidades de personagem principal são: objeto geométrico, automóvel, aeronave, animal, pessoa, etc. Os obstáculos podem ser objetos geométricos, automóveis, pedras/meteoros, chamas, etc.

O jogador tem uma quantidade predeterminada de chances (vidas). Ele deve controlar o personagem principal buscando evitar que esse colida com os obstáculos. Caso haja colisão, o jogo reinicia e o personagem perde uma vida. O jogo encerra quando o personagem não tiver mais vidas.

O jogo também deve fornecer ao jogador a possibilidade de escolher entre diferentes níveis de dificuldade. Os maiores níveis de dificuldades podem, por exemplo, aumentar o número de obstáculos ou a velocidade de movimentação deles.

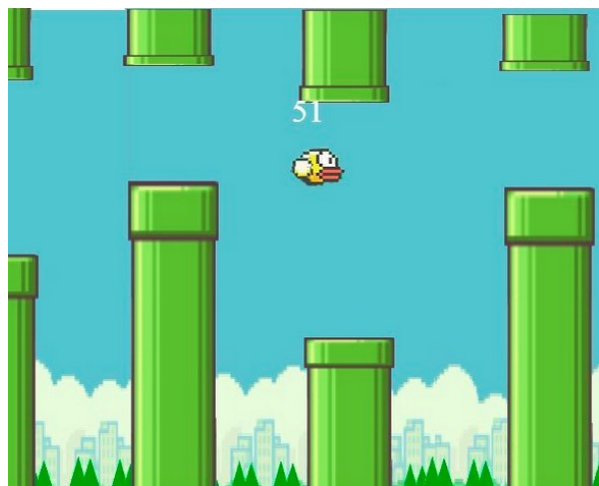
Muitos jogos digitais seguem esse conceito ao menos em alguma parte. Seguem alguns exemplos:



Jogos de Corrida



Jogos de Nave Espacial



Flappy Bird



Super Mario Bros

9. Links Úteis

- Tutorial Introdutório ao Processing: http://joaofaraco.com.br/arte_design/introducao-ao-processing/
- Especificação da Linguagem Processing: <https://processing.org/reference/>
- Especificação da Linguagem Java: <https://docs.oracle.com/javase/specs/jls/se8/html/index.html>