



Exercício Programa – Design de Software 2020.1 - “Craps Insper”

Data de entrega: **3/4/2020, via GitHub** / Grupos: **duplas**

Introdução

Neste Exercício-Programa vocês desenvolverão um jogo de Craps, que é um jogo de dados muito popular em cassinos. O jogo consiste em apostar no resultado de um par de dados. As regras apresentadas abaixo são simplificadas e não cobrem todas as possibilidades de um jogo real. As apostas serão sempre de números inteiros positivos de fichas, e o jogador começa com uma quantidade de fichas definida por você.

Regras

Os recursos básicos do jogo é feito por meio de rodadas sucessivas, onde o jogador deve decidir apostar ou sair do jogo, ou automaticamente sair se acabaram as fichas dele. Neste jogo uma rodada pode ter duas fases, começando com uma chamada de “Come Out” e conforme o resultado, passar para a fase de “Point”. O jogador pode fazer várias tipos de apostas conforme a fase. Deixe sempre o jogador informado de que fase ele está (detalhes a seguir). O computador é quem realiza o lançamento de dois dados (6 lados) para o jogo.

Conforme a fase do jogo, mostre para o jogador as possibilidades de apostas e pergunte qual a opção e o valor que deseja apostar, ele pode fazer apostas de mais de um tipo por vez. Os tipo de apostas são mostradas a seguir:

Pass Line Bet – Esta aposta só pode ser feita na fase de “Come Out”. Se a soma dos dados lançados for 7 ou 11 o jogador ganha (por exemplo: se apostou 10 fichas, mantém as 10 e recebe mais 10). Já se os dados somarem 2, 3 ou 12 (chamado de craps) o jogador perde (ou seja, se apostou 10 fichas, não recebe nada e perde essas 10). Já se a soma dos dados der 4, 5, 6, 8, 9 ou 10 o jogo muda para a fase de “Point” e o objetivo muda. A aposta já feita continua valendo, porém, o valor tirado se torna o Point e para o jogador ganhar agora, a soma do novo lançamento dos dados deve ser igual ao do Point. Se a nova soma dos dados é a mesma do que foi guardado no Point, o jogador ganha o mesmo valor que apostou. Se sair uma soma de valor 7 o jogador perde tudo. Caso saia qualquer outro número, se mantém na fase de “Point” sem perder ou ganhar e se continua lançando os dados até um veredito, quando sair ou o número do Point ou o 7, terminando essa rodada e deixando começar uma nova em “Come Out”.

Field – Esta aposta pode ser feita em qualquer fase do jogo. Nesta aposta se os dados derem 5, 6, 7 ou 8 o jogador perde tudo, já se derem 3, 4, 9, 10, ou 11 o jogador ganha o mesmo valor que apostou, já se a soma for 2 o jogador ganha o dobro do que apostou (se apostou 10 fichas, fica com as 10 e ganha mais 20), e finalmente se sai 12 nos dados o jogador ganha o triplo (se apostou 10 fichas, fica com as 10 e ganha mais 30).

Any Craps – Esta aposta pode ser feita em qualquer fase do jogo. Nesta aposta se o dados derem 2, 3 ou 12 o jogador ganha sete vezes o que apostou, senão perde a aposta.

Twelve – Esta aposta pode ser feita em qualquer fase do jogo. Nesta aposta se o dados derem 12 o jogador ganha trinta vezes o que apostou, senão perde a aposta.

Rubricas dos Objetivos de Aprendizado

Os objetivos de aprendizagem e as correspondentes rubricas de avaliação são os seguintes:

Objetivo: Desenvolver programas de computador.

I	D	C	B	A
Não fez nem ao menos os recursos básicos de sorteio	Fez os recursos básicos, porém mais nenhum.	Implementou corretamente os recursos básicos e pelo menos dois tipos de apostas.	Implementou corretamente os recursos básicos e pelo menos três tipos de apostas.	Implementou corretamente os recursos básicos e os quatro tipos de apostas.

Objetivo: Atuar em uma equipe gerenciada por métodos ágeis.

I	D	C	B / A
Não usou o git de forma correta.	Código entregue em um ou poucos <i>commits</i> . Caso em dupla não demonstraram que ambos os alunos contribuíram de modo substancial.	Vários recursos no mesmo <i>commit</i> . Caso em dupla: balanço de código desenvolvido é muito inclinado para um dos alunos.	<i>Commits</i> contínuos e caso em dupla balanço adequado de esforços.

Objetivo: Identificar e programar estratégias computacionais de resolução de problemas práticos.

I	D	C	B	A
Repositório não tem tudo para executar o programa.	Repositório com arquivos ou código desnecessário.	Código estruturado e documentado de forma confusa.	Código com estruturação e documentação razoável.	Código bem estruturado, modularizado e documentado.

A nota do projeto será dada pelo seguinte:

- Se a equipe obteve conceito I ou D em algum objetivo de aprendizagem, o conceito final será dado pelo menor conceito obtido. Por exemplo: se todos os recursos foram implementados, e o código está impecável, mas um aluno fez tudo (conforme evidenciado no git), a nota será D ou I (Não adianta argumentar que foi feito tudo no computador de um aluno só, o objetivo aqui é que todos saibam trabalhar de forma colaborativa usando uma ferramenta como o git).
- Se todos os conceitos forem iguais ou superiores a C, o conceito final será obtido pela média aritmética dos conceitos obtidos para os objetivos de aprendizagem.

Referências

Git – No Blackboard da disciplina existe um documento que ensina como usar o básico do Git (que é o que é necessário agora).

Craps – Existem diversos guias de Craps na internet, contudo não há a necessidade de implementar o jogo completo que é bem complexo, mas se houve curiosidade, esse site tem um simulador online interessante: <https://wizardofodds.com/play/craps/v2/>

Entrega

Envie no BlackBoard o endereço do seu repositório GitHub. Exercícios copiados ou muito parecidos podem gerar notificações de infrações do código de ética Inspêr, resultando em reprovação automática da disciplina ou até mesmo jubramento.