





A **Amongsancle Enunciado** 
Project ID: 4404

Forked from an inaccessible project.



Actualizar README.md

José Lozano Barreiro authored 1 week ago

Name	Last commit	Last update
 video	video actualizado	1 week ago
 README.md	Actualizar README.md	1 week ago
 image.png	Primer commit	1 week ago

[README.md](#)

Amongsancle Enunciado



Descrición xeral

Se trata de implementar unha simulación do coñecido xogo Among Us. Esta práctica está baseada a súa vez na práctica que fixo no 2022 o profesor de prácticas Alvaro Ordoñez.

En *Among SanClemente*, un grupo de estudantes quedou encerrado nas instalacións do IES San Clemente e debe realizar todas as tarefas encomendadas para poder continuar coa actividade normal do centro. Cambiar un cable *Ethernet* na sala de profesorado, ir coller xices á conserxería para repoñelos nas aulas ou reparar o proxector na aula da bodega son só algunhas das duras tarefas que o estudantado terá que solucionar. Pero isto non é todo ao que se terán que enfrontar! Entre eles/as hai algúns/has impostores/as que farán todo o posible para evitar que o grupo de estudantes consiga realizar todas as tarefas e recuperar o transcurso habitual do IES San Clemente.

O programa que imos desenvolver simulará o comportamento de todos os/as xogadores/as (estudantes e impostores/as) e o obxectivo da

cada rolda, os/as xogadores/as (estudantes e impostores/as) realizarán virtualmente as tarefas encomendadas. As tarefas estarán almacenadas nunha cola e en cada rolda terán que realizar a que estea en primeira posición (cola é FIFO). O programa dirá quen fixo que tarefa e en que localización. Tamén avisará de se se produciu algunha eliminación por parte dun/ha impostor/a. Todo isto será automático pero serás ti, usuaria, a persoa que ten que acusar en cada rolda (ou pasar para que continúe o xogo) a un/ha posible impostor/a cos datos que che dá o programa.

O xogo finalizará cando:

- os/as xogadores/as estudantes terminen todas as súas tarefas (vitoria).
- se expulsen a todos os/as impostores/as (vitoria).
- o número de impostores/as sexa maior ou igual que o número de estudantes (derrota).

Máis descrições e restriccións que debe cumprir a tua App

- O proxecto debe estar en VSC por si eu quixera/precisara clonalo cómodamente coa opción específica do gitlab.
- Uso de paquetes. Nos despreocupamos nesta práctica da estrutura de paquetes. Poden estar tódalas clases no paquete por defecto.
- Evita static sen sentido. Unha cousa e "utilizar ben" static por exemplo no patrón Singleton e outra facer un main que chama a un porrón de métodos static. De prácticas anteriores xa ten que estar asimilada esta cuestión.
- Procura que clases elementales como Xogador, Tarefa etc. sexan independentes da E/S pero para o resto do código se consinte nesta práctica mezclar lóxica con E/S co obxectivo de facer o código máis curto e centrar o noso esforzo na xestión das listas.
- Haberá dous tipos de xogadores/as: estudantes e impostores/as.
- Atributos comúns a estudantes e impostares: un alias e unha cola de tarefas.
- Atributos só das/os impostores/as: lista onde se almacenen os/as estudantes que eliminaron.
- Hay que comprobar que non se introducen alias repetidos a hora de crear novos/as xogadores/as e ** **que os alias teñen o formato @xxx donde cada x pode ser calquera letra ou díxito. observa que tódolos alias comenzan con @
- Cada vez que se crea un xogador, asígnaselle a devandito xogador/a unha lista de tarefas ** *(5 ou mais) ** **escollidas aleatoriamente da lista xeral de tarefas
- Polo tanto, tes que definir unha lista xeral de tarefas. Unha tarefa ten 2 atributos: nome(descripción) e habitación. O nome sería por exemplo "Arranxar o proxector" e a habitación "Aula da bodega". Procurade que haxa máis dunha tarefa na mesma habitación.
- Antes de iniciar unha partida(opción xogar no menú), o programa de xeito aleatorio configura que estudantes pasan a ser impostores durante o transcurso da partida. O número de impostores será ≥ 1 e o número de xogadores totales(estudantes + impostores) lóxicamente ten que ser maior co número de impostores
- Os listados serán sempre ordenados, cando o vexas conviente podes ordenar por varios campos, por exemplo, o listado de tarefas pode de ser ordenadas por habitación e para a mesma habitación por descrición.
- Como se indicou anteriormente a partida se organiza en roldas. En cada rolda cada impostor observará os xogadores que están na sua habitación e intentará asesinalos O intento se simula a través dunha decisión random tipo cara o cruz
- Pode haber varios asasinatos na mesma rolda
- Os impostores non se poden matar entre eles.
- Sobre o control de tempo de reposta do usuario para indicar a quen quere expulsar.

Durante o transcurso da partida o usuario debe indicar quen é o impostor expulsando ao xogador do que sospeita. O programa debe controlar o tempo que tarda en responder o usuario e cotexalo cun tempo límite de resposta. O tempo límite de resposta se configura coa opción correspondente de menú. Si o usuario supera o tempo límite de resposta non se lle ten enconta o que escriba e non pode expulsar a ningún na rolda actual. Si un usuario responde fora do tempo límite, incluso se pode sancionar ao usuario reducíndolle o tempo de resposta nas próximas roldas para estresalo e evitar que analice a información tranquilamente.

- unha vez terminado o xogo, poderase volver a xogar sin salir da APP. Habrá que restaurar os xogadores ao estado inicial de forma que todos sexan estudantes vivos e sin expulsar pero poden manter as tarefas asignadas. Tamén para o novo xogo e posible reconfigurar os xogadores, tarefas e tempo de resposta vía menú.
- Tras rematar a partida non temos que saír necesariamente do programa. O programa permitirá xogar unha nova partida cos mesmos usuarios. e coas mesmas tarefas asignadas que na partida anterior ou por suposto configurar as tarefas e xogadores a través do menú antes de comenazar a nova partida.

PATRÓNS

asociado a System.in. Ao ser unha App pequena e feita por un único programador sería suficiente con a referencia ao teclado fora unha simple variable compartida por todas as clases pero por motivos didácticos nos forzamos a resolver esta cuestión con Singleton.

- O menu utiliza o patrón composite e se estrutura da seguinte maneira.

MenuAmongSancler:

0. Configuración

0. Tarefas

0. Engadir tarefa

1. Borrar tarefa

2. Ver tarefas.

3. Salir

1. Xogadores.

0. Engadir xogador

1. Borrar xogador

2. Ver xogadores

3. Salir.

2. Configurar tempo máximo de resposta

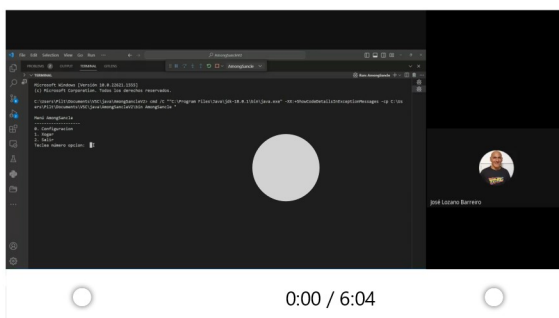
3. Salir.

1. Xogar

2. Salir

Para o menú anterior xa sale a conta aplicar o patrón composite. Pensa ademais que ata para o noso sinxelo suposto o menú anterior pode requirir ampliacións e reestructuración rápidas o cal confirmaría todavía máis a necesidade do patrón.

un pequeno video



[among.mp4](#)

Melloras e ampliacións

Acéptanse todo tipo de melloras e ampliacións.

ENTREGASE

- na tarefa de moodle a url do teu proxecto gitlab.
- proxecto gitlab.Non olvides incluír como membro a lozano@iessanclemente.net

O proxecto gitlab debe ser proxecto VSC que debe conter

- Código java

GitLab IES San Clemente

- Estructura do menú, navegando rápido por todas as opcións posibles sin pararse na execución das follas.
 - 2 partidas consecutivas(sin sair de APP) rápidas para non facer extenso o video. É imposible unha descripción detallada da tua App nun video de 5-10min. Centráte simplemente nos dous puntos indicados. Non markdown que se pide máis abaixo poderás si o desexas incorporar calquera explicación ou observación.
- Documento markdown con:
 - Impresións ao escribir a App. ¿tiveches nalgún momento a sensación que algunha parte do código a estabas escribindo de forma non apropiada por falta de non aplicar un patrón non visto ou por calquera outra cuestión?.
 - Indica si precisañes empregar iteradores donde e porqué.
 - Indica si precisañes sobreescribir hashcode>equals donde e porqué.
 - Indica donde usaches comparable/comparator e porqué.
 - Unha lista breve das meioras achegadas por ti, e dicir, de funcións e restriccións que non se piden no enunciado.
 - Outras observacións.