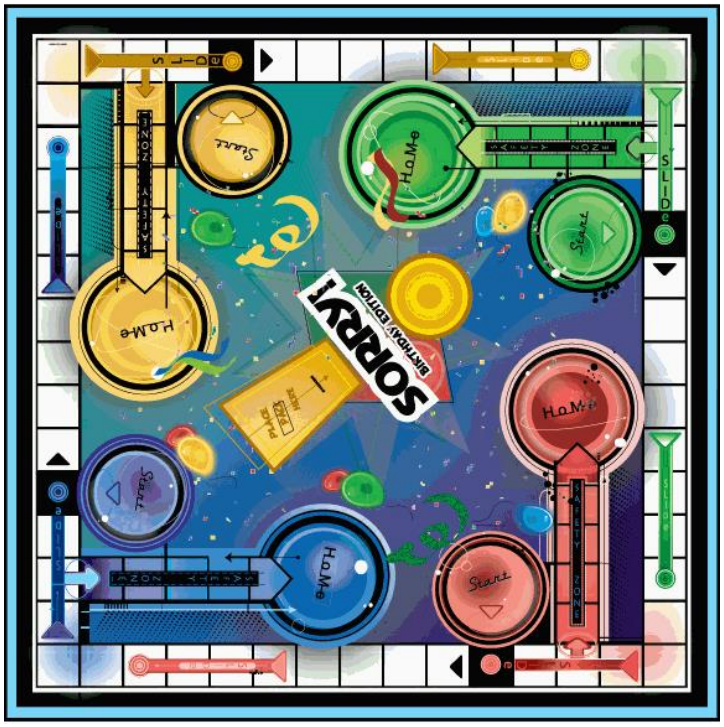
Universidad del Valle de Guatemala
Programación Orientada a Objetos, CC2006
Sección 10
Catedrático: Tomás Gálvez P.

Ejercicio sobre principios de programación con orientación a objetos

Implemente, en un programa de Java, el juego *Sorry!* diseñado en su diagrama UML bueno del ejercicio 7. Agregue a su diseño y su programa lo necesario para implementar características adicionales de las reglas clásicas (de acuerdo con Wikipedia y WikiHow). Elija las características que desarrollará tomando en cuenta el valor de cada una:

- **(30 pts.)** Límite de cartas: el juego presenta 45 cartas en total; cinco unos y cuatro de cada una de las demás cartas (incluyendo las cartas SORRY!). Esto se refleja en las probabilidades de que aparezca cada carta conforme el juego avanza. Esto se puede implementar generando un banco de cartas que se baraja cuando es agotado, o con probabilidades cambiantes por carta.
- **(30 pts.)** *Bots*: el juego permite seleccionar *bots* para llenar los espacios de jugadores. Debe haber al menos un jugador humano.
- **(20 pts.)** *Slides* sobre el tablero: cada color tiene dos espacios en su "lado" (desde su posición de inicio hasta la posición de inicio del "siguiente" color) tales que los peones que caen en ellos son adelantados cuatro espacios automáticamente. La ubicación de los espacios para cada color puede apreciarse en la siguiente imagen:



- **(20 pts.)** Contraataques: se añaden ocho cartas a la categoría de cartas SORRY!; cuatro cartas para que un jugador evite que su peón sea enviado a su área de inicio; y cuatro cartas para contrarrestar esta defensa. Al ser todas cartas de tipo SORRY!, estas cartas están sujetas a las reglas establecidas para posesión de cartas SORRY!
- **(10 pts.)** Retrocesos: la carta 4 siempre retrocede. Si un jugador saca la carta 10, debe avanzar 10 espacios o retroceder 1.

- **(10 pts.)** Distribución: la carta 7 debe distribuir su siete avances entre dos peones fuera del área de inicio (se permite que un peón se mueva cero espacios y el otro siete).
- **(10 pts.)** Zonas seguras: los últimos 5 espacios antes de llegar a la *home* de cada color son accesibles únicamente para el color correspondiente, y son la zona segura para los peones de ese color. Los peones en la zona segura son inmunes a regresos al área de inicio y a intercambios de posición.
- **(10 pts.)** Tablero "circular": si un peón retrocede lo suficiente desde su área de inicio, podrá desviarse en camino a su *home* al avanzar.
- **(5 pts.)** Múltiples cartas SORRY!: los jugadores pueden tener más de una carta SORRY! en su posesión.
- **(5 pts.)** Inicio: los jugadores compiten al comenzar el juego para determinar quién saca la primera carta.
- **(5 pts.)** Intercambios: si un jugador saca 11 puede avanzar 11 espacios o intercambiar su posición con la del peón de otro jugador que esté fuera de su área de inicio.
- **(5 pts.)** Salidas: sólo se pueden sacar peones del inicio cuando se saca carta con un 1 o un 2.
- **(5 pts.)** Segunda carta: si un jugador saca la carta 2 deberá tomar una carta adicional.

Debe entregar en Canvas:

- **[60%]** Cambios al diagrama de clases incluyendo las características a implementar, elegidas por el desarrollador.
- **[40%]** Implementación del juego incluyendo los cambios hecho al diagrama de clases, de acuerdo con el punto anterior.

Tome en cuenta las responsabilidades y su distribución en el diagrama UML que hizo originalmente. Considere que los cambios que aplicará para incluir las características de este laboratorio podrían requerir cambios a las responsabilidades e incluso la generación de más clases. Asegúrese de adherirse a la Ley de Demeter y a los principios única responsabilidad y alta cohesión/bajo acoplamiento.

El porcentaje de punteo indicado es por característica. Es decir, el 60% del valor de cada característica se basará en cómo la característica es agregada al diagrama de clases; y 40% se basará en la implementación de dicha característica como parte del programa.

La nota máxima que podrá obtenerse en esta actividad es 100 pts.