Taller 3 Diseño

Iteración 1:

Se determinó como rol el dominio general del juego y se definieron los requerimientos funcionales

Roles: Pac-Man

Responsabilidades:

1. Iniciar juego
2. Mover pac-man
3. Cargar tablero
4. Cargar frutas
5. Cargar galletas
6. Modo Invencible
7. Comer fantasma
8. Jugador muere
9. Ver record
10. Actualizar record
11. Mover fantasmas
12. Agregar fantasmas
13. Set personalidad
14. Cambiar nivel de juego

Iteración 2:

Se definieron conceptos que cumplen o son parte de las responsabilidades con sus debidos estereotipos y descripción

* Jugador: Information Holder

Es el usuario con sus artributos, vidas y record

* Fantasma: Information Holder

Tiene personalidad y usa a CalculadoraMovimiento

* Galleta: Information Holder

Número que corresponde al tamaño de matriz – número de frutas, el juego acaba cuando llegan a 0, las galletas se reducen cuando un jugador de desplaza

* Frutas: Information Holder

Número que corresponde a los poderes del jugador

* Tablero: Structurer

Estructura que se imprime en cada iteración

* Record: Information Holder

Guarda 10 nombres de los mejores puntajes

* CalculadoraMovimiento: Controller

Determina a dónde se mueve un fantasma según su color, es decir, maneja los algoritmos de cada fantasma

* ManejadorTablero: Controller

Pone las frutas y galletas en el tablero

* Admin: Controllador

Calcula los puntos del Jugador al avanzar por cada Casilla en base al contenido de la misma o al tener la misma posición que un fantasma siendo el Jugador invencible

Diagrama de clases

Diagrama

Descripción generada automáticamente

Reflexión

Ventajas:

* Se pudo relacionar adecuadamente los conceptos y se redujo el número de conceptos y responsabilidades, pasando de 14 a 9.
* Al definir los conceptos, se reflejó la relación entre clases
* Los estereotipos de los conceptos nos permitieron limitar los métodos para que un information holder no afecte a otra clase, pues dejaría de ser information holder
* Como cada personalidad de un fantasma es una sub-clase, al crear un nuevo fantasma requiere más información y trabajo. Caso contrario cuando la personalidad es un atributo y la clase Juego calcula algoritmos en base a dicho atributo del Fantasma, siendo muy extensas las responsabilidades de la clase Juego.

Desventajas:

* Las clases Fantasma y Jugador tienen atributos de posición en 2 dimensiones, esto puede generar tiempo extra de ejecución al sobrescribir los datos continuamente, lo cual sería diferente en el juego de serpientes y escaleras que puede ser representado linealmente.

Trade-offs:

* Los conceptos de CalculadoraMovimiento, ManejadorTablero y Admin se incluyeron en la clase Juego, lo cual reduce el número de clases pero incrementa la eficiencia al tener varias responsabilidades en diferentes clases. Mientras que, al haber tenido 3 clases diferentes, cada una pudo hacer tareas específicas en menor tiempo al tener menos tareas.
* La clase Record pudo haber sido un atributo de Juego sin embargo, en el diseño se evidenció la importancia de la relación entre Jugador y Record, por lo que esta última se convirtió en una clase aparte.