**Instituto Tecnológico de Costa Rica**

**Ingeniería en Computación**

**Curso IC-1803 Taller de programación**

**Descripción del primer proyecto programado**

**“Snake”**

**Valor: 30%**

**Profesores:**

**Dennis Valverde**

**Efrén Jiménez**

**Vera Gamboa**

**-I semestre 2017-**

# **Introducción**

Este documento presenta la propuesta del segundo proyecto programado del curso Taller de Programación a ser desarrollado por los estudiantes de la Carrera Ingeniería en Computación, en el I semestre del 2017

# **Objetivo general**

Desarrollar un proyecto de complejidad media que permita aplicar conocimientos básicos de orientación a objetos.

# **Objetivos específicos**

* Escribir funciones que permita la interacción usuario-computadora para alimentar un sistema computacional.
* Utilizar estructuras tipo listas y diccionarios e archivos para guardar y recuperar información
* Utilizar paradigma Programación Orientada a Objetos.
* Construir un programa con los elementos básicos de un ambiente de programación.
* Diseñar, Implementar, validar y documentar programas.

# **Descripción**

Se desarrollará una aplicación en Python que permita a replicar el video juego “Snake”. En el juego, el jugador o usuario controla una larga y delgada criatura, semejante a una serpiente, que vaga alrededor de un plano delimitado, recogiendo alimentos (o algún otro elemento), tratando de evitar golpear a su propia cola o las "paredes" que rodean el área de juego. Cada vez que la serpiente se come un pedazo de comida, la cola crece más, provocando que aumente la dificultad del juego. El usuario controla la dirección de la cabeza de la serpiente (arriba, abajo, izquierda o derecha) y el cuerpo de la serpiente la sigue. Además, el jugador no puede detener el movimiento de la serpiente, mientras que el juego está en marcha.

**Usuario**

* El usuario podrá crear nuevas partidas, consultar máximas puntuaciones y también configurar el juego para aumentar o disminuir la dificultad del mismo.
* Se debe almacenar todas las partidas del usuario.
* Cada vez que inicia sesión un usuario debe escoger alguna partida guarda o iniciar una nueva partida.
* Seguido debe aparecer el menú correspondiente.

## **Registro y login:**

Como medida de seguridad para ingresar a la aplicación los usuarios deben de estar registrados previamente

El primer menú de la aplicación debe permitir al usuario hacer login con su respectiva contraseña de acceso, en caso de ya estar registrado. En caso contrario se debe de crear una cuenta para registrarse

* Para registrarse es necesario almacenar:
  + **usuario, correo electrónico y contraseña** de cada usuario. El **ID** de cada usuario será autoincremental.
* Todos los usuarios deben estar registrados en la misma lista.

## **Partidas:**

* Nueva partida, el usuario podrá crear nuevas partidas y escoger el nivel que desea jugar
* Continuar partida, el usuario podrá salirse de una partida y dejar esta para jugar luego.
* Eliminar partida, se le mostrara al usuario todas las partidas que no han sido terminadas y luego, se escogerá cuales se desean eliminar.
* Salir partida, el usuario podrá salir de la partida en el momento que desee.

## **Configuración:**

* Se podrá modificar los valores del primer nivel.
* Se podrá modificar el signo usado para representar el Snake.
* El valor inicial del terreno de juego de es 5 filas y 5 columnas

## **Bloques:**

* Existen 5 tipos de bloques Vida, Obstáculo, Comida, Bonus, Bonus especial
* Los bloques de tipo vida, solo existirán uno por nivel y se colocara aleatoriamente.
* Los bloques de tipo de obstáculo cuando son sobrepasados por el snake, hará que se pierda una vida, hasta un máximo de 3 vidas.
* El bloque de comida hará, que el Snake crezca lo cual debe evitar que el tamaño total del snake no sobrepase algún obstáculo o línea del plano de juego.
* El bloque de Bonus deberá aparecer aleatoriamente cada 2 niveles y brindará un bloque de comida y un bloque de vida
* El bloque de Bonus especial deberá aparecer aleatoriamente cada 5 niveles y brindará un nuevo nivel.

## **Mecánica del juego:**

* El Snake puede cambiar de dirección de arriba o abajo, de izquierda a derecha siempre y cuando la pieza inicial permita dicho movimiento.
* Cada vez que el snake encuentre un bloque de comida esta crecerá hasta que suba de nivel.
* El juego se acaba cuando la snake se topa con el límite del rectángulo de juego.
* Cada nivel tiene un máximo de 3 vidas por nivel, luego de perderlas se acabará la partida.
* Cada vida se perderá cuando se encuentre un bloque de obstáculo, o se toque alguna de las 4 esquinas.
* Los bloques de comida son colocados aleatoriamente en el plano y tienen un límite de movimientos antes de desaparecer.
* Los bloques de obstáculo se colocan aleatoriamente y se mueven cada 3 movimientos aleatoriamente también.
* El nivel inicial contiene una matriz de 5 x 5 y sube al siguiente nivel con 1 bloque de comida, un bloque de obstáculo
* Cada vez que se sube de nivel se exponen al cuadrado los bloques y se suma a la matriz un nueva columna y fila.

**Plano 5x5**

**Vidas \*\*\***

**Nivel 1**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
|  | Obstáculo |  |  | Comida |
|  |  | Snake |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

**Plano 6x6**

**Vidas \*\*\***

**Nivel 2**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
|  | Obstáculo |  |  |  |  |
|  |  | Snake |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  | Obstáculo |  |
| Comida |  |  |  |  |  |

## **Estadísticas:**

* EL usuario podrá ver sus 10 mejores partidas y sus puntuaciones
* EL usuario podrá ver el top 5 de los mejores jugadores con sus puntuaciones
* EL usuario podrá ver el promedio de puntos por las partidas jugadas
* EL usuario podrá ver la cantidad de partidas guardas, eliminadas.
* EL usuario podrá ver el nivel máximo y puntuación obtenida.

## **Aspectos importantes**

* El sistema deberá ser ejecutado en consola, opcionales componentes gráficos. Preguntar al profesor por esta opción.
* El uso de archivos para guardar la información es opcional. Preguntar al profesor por esta opción.
* Tome en cuenta la explicación del profesor en clase al momento de entregar este enunciado.
* Las estructuras permitidas en este proyecto son listas y diccionarios. Cualquier otra estructura debe ser aprobada por el profesor.
* Cualquier requerimiento del sistema que no se encuentre descrito en este enunciado y que el estudiante considere que sea indispensable para el desarrollo de la solución debe ser comentado con el profesor para su aprobación.

## **Requerimientos funcionales:**

* Cada bloque existente en el plano de juego, debe respetarse.
* Cada bloque asignado en el plano debe contener un objeto del tipo correspondiente.

## **Documentación**

Se deberá documentar cada uno de los métodos desarrollados para obtener la solución del sistema. A esto se le conoce como documentación interna y se hace mediante comentarios dentro del código Python.

Se deberá generar adicionalmente una documentación externa del proyecto en formato PDF con la estructura definida en el archivo “Estándar de documentación externa”, disponible en la carpeta “Proyectos” en el Tec-Digital.

Para que la documentación sea válida cada estudiante debe haber confeccionado al menos el 40% de ésta, de otro modo su nota es 0. La participación de cada estudiante se indicará en la bitácora de participación.

## **Evaluación**

Si bien es cierto el proyecto puede ser grupal, la nota será individual, esta se calculará con una formula basada en el porcentaje de participación del estudiante en el proyecto, este dato se comprobará mediante una entrevista técnica que el profesor realizará a cada uno de los estudiantes basándose en la bitácora de participación que estos le brindarán el día de la revisión.

La nota preliminar del proyecto se evaluará de la siguiente forma: a ajustar

|  |  |
| --- | --- |
| **Aspecto** | **Valor** |
| Navegación por medio de menús | 10% |
| Login y registro | 10% |
| Mecánica del juego | 45% |
| Estadísticas | 10% |
| Uso de los bloques |  |
| Documentación | 15% |

Una vez calculada la nota preliminar, la nota final del proyecto se calculará de la siguiente forma:

* Si se comprueba mediante la entrevista que cada estudiante participó al menos en un 47.5% de los puntos obtenidos en la nota preliminar, esta se convertirá en su nota final.
* Si se comprueba mediante la entrevista que un estudiante participó en MENOS de un 47.5% de los puntos obtenidos en la nota preliminar se aplicará la siguiente fórmula:

Nota final =

**Bitácora de participación**

Deberá ser traída el día de la revisión. Contiene el porcentaje de participación de cada estudiante en cada una de las tareas. El profesor mediante una entrevista comprobará que los datos suministrados sean reales y sino basado en esta asignará los porcentajes demostrados por el estudiante.

A continuación, un ejemplo de la misma:

|  |  |  |
| --- | --- | --- |
| **Aspecto** | **María** | **José** |
| Navegación por medio de menús | 40% | 60% |
| Mecánica del juego | 70% | 30% |
| Estadísticas | 60% | 40% |
| Uso de los bloques | 50% | 50% |
| Documentación | 40% | 60% |

**Aspectos administrativos**

* La tarea será desarrollada en grupos de trabajo de máximo 2 personas. Cualquier acción de plagio será sancionada con un **0** en la nota y la carta al expediente respectiva para todos los implicados.
* La fecha de entrega será el viernes 2 de junio de 2017 a las 23:55.
* Se entregarán informes de avance vía email los días 11 y 18 y 25 de abril de 2017.
* La documentación externa, (interna, informes de avance, código y menús del sistema deberán ser redactados en inglés.)
* La entrega será mediante la plataforma Tec-Digital. Se emplearán citas de revisión para calificar cada proyecto.
* Cualquier archivo infectado con virus no será revisado.
* Se recomienda que se empiece a trabajar lo más antes posible.