**Propuesta de desarrollo de juego**

**como entrega final de Infomática II:**

**Gravity Bike**

**Estudiante:**

**Juan Mauricio Parrado Villa**

**Materia:**

**Informática II**

**Universidad de Antioquia**

**Facultad de Ingeniería**

**Departamento de Electrónica y Telecomunicaciones**

**Descripción de su proyecto:**

El juego a desarrollar se basa en la práctica de algunos jóvenes en ciudades principales de Colombia que realizan en las calles principales con pendiente suficiente, donde usan bicicletas pequeñas adaptadas para alcanzar grandes velocidades. Dicha práctica ha adquirido el término de “**Gravity Bike**” o “Descuelgüe” y ha sido popular durante los últimos años entre los jóvenes menores de edad, por su alto grado de riesgo y la alta descarga de adrenalina que esto genera.

Este juego pretende emular dicha práctica, incluyendo los riesgos reales y aplicando leyes de newton para emular el movimiento, coeficientes de fricción para el frenado. El juego tendría obstáculos respectivos a su contexto: carros, motos, buses, postes, huecos, etc. Y su nivel de dificultad depende de dicho número de obstáculos.

**Justificación:**

Como se enuncia en la descripción, esta práctica es realizada en su mayoría por jóvenes menores de edad en las ciudades principales de Colombia y conlleva un algo riesgo ya que debido a su informalidad, los jóvenes no cuentan con la protección necesaria para minimizar los accidentes.

El juego está diseñado para generar una sensación similar a la práctica real pero con un riesgo nulo, ofreciendo diversión sin poner en riesgo tu vida y a su vez muestra un contexto similar a la práctica real, es decir las calles de Medellín donde se suele practicar esta actividad.

**Motivación:**

Esta idea surgió de la observación de dicha práctica en la ciudad de Medellín, la cual lamentablemente ha tenido consecuencias mortales para muchos jóvenes en la ciudad. Sin embargo, como ciclista entiendo la sensación de adrenalina que se tiene al descender a grandes velocidades en calles de la ciudad, con la protección adecuada.

Con esto en mente quise crear un juego que permita dar una idea de dicha sensación si poner en riesgo tu vida.

**Desafíos a afrontar:**

Para el desarrollo de este juego se pretende usar superposición de imágenes en dos dimensiones para simular un efecto 3D y darle profundidad al juego; este sería el mayor desafío en términos de desarrollo, a su vez la creación de los diferentes escenarios donde se llevará a cabo las carreras, ya que se tiene planteado realizarlos en calles de Medellín así que conlleva un reto en diseño.

Por otro lado, en términos del modelo de descripción física, se tendrán en cuenta aspectos como caída libre, resistencia al aire y fricción con el suelo, así como colisiones de objetos macroscópicos.

**Descripción de las partes:**

Respecto al juego tendrá las siguientes partes:

* Pantalla de inicio: En esta sección el jugador o jugadores tendrán la opción de login, dando un usuario y contraseña con el cual podrán ingresar a los juegos guardados. También estará la opción para crear una cuenta en caso de que no se cuente con una anteriormente.
* Selección de personaje: En esta sección el jugador tendrá acceso a seleccionar el personaje con el que desea jugar.
* Sección del Juego, corresponde al módulo donde se desarrolla el juego y el jugador tendrá acceso a interactuar con el personaje seleccionado anteriormente en los distintos escenarios.
* Menú: cuando se pause el juego, el jugador tendrá la opción de guardar el juego en todo momento y de reanudar partidas anteriormente guardadas, así como salir del juego o volver a la pantalla de inicio.

**Diagrama de clases:**

Estas son las clases propuestas para la construcción del proyecto.

* **Background:**

Esta clase contendría los distintos fondos de cada sección del juego, inicio, selección del personaje, desarrollo del juego, pausa y cambiará de acuerdo con dicho menú y a los niveles del juego.

La clase contendría los siguientes atributos:

* + **Image:** corresponde al nombre del archivo de la imagen que se encuentra cargada actualmente como fondo.
  + **Type:** corresponde al tipo de sección del programa en el que el usuario se encuentra actualmente, puede variar en los siguientes valores:
    - Main\_Menu
    - Character\_selection
    - Road\_selection
    - Game
    - Paused

La clase contendrá los siguientes métodos:

* + **Set\_image:** selecciona la imagen de fondo que corresponde a la sección que se encuentre en ejecución.
  + **Set\_type:** le da valor a la variable type de acuerdo con la sección del juego en la que se encuentre el usuario.
* **Character:** esta clase corresponde al personaje que se mueve en la sección de juego, interactúa con el usuario con las flechas del teclado.

El personaje se encontrará en todo momento rodando en bicicleta con fricción en el suelo debido a un movimiento de rodar sin deslizar, fricción con el aire debido a forma que cambiará de acuerdo con posición del personaje en la bicicleta.

Esta clase tendrá los siguientes atributos:

* + **X\_Velocity:** (double) esta variable estará definida por la velocidad en x del personaje con respecto al suelo en reposo.
  + **Y\_Velocity:** (double) esta variable estará definida por la velocidad en y del personaje con respecto al suelo en reposo.
  + **X\_acceleration:** (double) esta variable estará definida por la aceleración en X del personaje con respecto al suelo.
  + **Y\_acceleration:** (double) esta variable estará definida por la aceleración en Y del personaje con respecto al suelo.
  + **X\_Position:** (double) esta variable estará definida por la posición en X del personaje, respecto a un punto de referencia en el mapa del juego.
  + **Y\_Position:** (double) esta variable estará definida por la posición en y del personaje, respecto a un punto de referencia en el mapa del juego.
  + **On\_floor:** (boolean) esta variable define si el personaje se encuentra rodando en el suelo, o suspendido en el aire debido a sus movimientos.
  + **C\_position:** (string) variable que define el tipo de posición que se encuentra el personaje en la bicicleta, sus opciones son “normal”, “superman”, “wheelie”, “breaker”.

La clase tendrá los siguientes métodos:

* + **Set\_x\_velocity:** método que le da valor al atributo x\_velocity.
  + **Set\_y\_velocity:** método que le da valor al atributo y\_velocity.
  + **Set\_x\_acceleration:** método que le da valor al atributo x\_acceleration.
  + **Set\_y\_acceleration:** método que le da valor al atributo y\_acceleration.
  + **Move:** método del tipo public slot que permite la ejecución de movimientos del personaje o personajes a través de las fechas del teclado para esquivar obstáculos y realizar trucos.
* **Obstacles:** Esta clase corresponde a distintos elementos que estarán dentro del mapa que serán obstáculos en el movimiento del personaje, la forma es de un objeto rectilíneo y el tamaño de los objetos son determinados en cada mapa y el número de objetos dependen del nivel de dificultad.

Esta clase tendrá los siguientes atributos:

* + **Image:** corresponde al nombre de archivo de la imagen que será mostrada en el juego.
  + **Width:** corresponde al ancho del objeto.
  + **Height:** corresponde la altura del objeto.
  + **Type:** corresponde al tipo de obstáculo, puede ser “hard”, “soft”, siendo hard el que finaliza el juego con una colisión, mientras que el soft cambia la respuesta del personaje a los movimientos dados por el usuario con un nivel de retraso.
* **Road:**

Corresponde a una serie de objetos tipo **QGraphicsItem** en conjunto con un path y una curva cúbica de Beizer, que determinan la forma de la calle que recorre el personaje en bicicleta.

**Calendario:**

* **Semana 1:** ~~Planeación y creación de propuesta.~~
  + ~~Idea Inicial~~
  + ~~Diagrama de clases~~
  + ~~Cronograma~~
  + ~~Evaluación de factibilidad~~
* **Semana 2:** Implementación en QT-C++:
  + ~~Diseño de fondo, personaje y obstáculos~~
  + Creación de clases y objetos
    - Creación de ~~clases~~, métodos y atributos
    - Creación de objetos
  + ~~Implementación de fenómenos físicos~~
    - ~~Rodar sin deslizar~~
    - Fricción al frenar
    - Torque
  + Creación de 3 mundos:
    - 1er mundo
    - 2ndo mundo
    - 3er mundo
  + Implementación 3 niveles de dificultad
  + Implementación multijugador
  + Implementación de gestión de partidas y acceso
  + Creación de menús de bienvenida y pausa
* **Semana 3:** Pruebas y despliegue:
  + Pruebas de estabilidad.
  + Pruebas de dificultad y jugabilidad
  + Pruebas de control de errores
  + Pruebas de ejecución de menús
  + Despliegue como archivo ejecutable

**Laboratorio número:** Grupo 5