**Drunk guy**

**Informática II**

**Juan Sebastián Marín Valencia**

**Felipe Macías Rodríguez**

**Departamento de Ingeniería Electrónica y**

**Telecomunicaciones**

**Universidad de Antioquia**

**Medellín**

**Introducción**

Este juego será, como su nombre lo dice, acerca de una persona borracha y de algunas situaciones que puede vivir o experimentar una persona bajo estado de embriaguez, presentándolas de forma bizarra y exagerada. Sin embargo, sin olvidarnos que esto es un juego y tampoco estará buscando netamente la fidelidad con la realidad. Ahora bien, ¿Por qué un juego con esta primicia? Si pensamos en todas las posibles situaciones que le pueden pasar a un borracho y lo divertidas que podrían ser, esta pregunta tendría una respuesta muy obvia y es que fácilmente, hasta una ida al baño es un desafío cuando estas en condiciones de ebriedad. Pero no solo nos tenemos que quedar con esto, ¿cómo sería manejar un carro? ¿pelear? Y un sinfín de situaciones que, aunque puedan ser cotidianas, a la hora de estar borrachos tienen un factor diferencial.

De eso se trata el juego, de que existan diferentes niveles u eventos donde haya un objetivo, como por ejemplo, un evento en un baño , el objetivo sería poder apuntar el chorro de orina en el inodoro; otro evento en la carretera intentando no chocar en el carro; otro en la casa intentando no pegarle a la esposa(recalco acá el tema de que son situaciones bizarras y exageradas) entre otros eventos que se pueden presentar, pero esto con el agravante de que no tienes el control total de lo que está pasando, pues estas borracho, es por esto que el factor fundamental en el desafío del juego es la aleatoriedad del personaje que estas tratando de manejar.

La puntuación dependerá en el nivel que estes. Si estas por ejemplo en el nivel del baño, la puntuación dependerá de que porcentaje de orina metiste en el inodoro. Tendrá 3 estrellas cada nivel (con una se habrá superado, con 3 se habrá hecho perfecto). Planea ser un juego casual que, si bien su ***gameplay*** puede parecer sencillo, la variedad en los niveles, las situaciones y los chistes en cada nivel que le metamos al juego ambientando esa sensación de que definitivamente se está manejando a un borracho, hacen del juego algo con potencial para entretener y sacar un buen rato.

**Niveles, vistas y físicas**

* Tiro al blanco

Este nivel consiste, como se mencionó en la introducción, en la persona borracha orinando, con el objetivo de no derramar orina afuera del sanitario. Además, a medida que avanza el tiempo el área del sanitario se va disminuyendo. Sin olvidar el factor de que la persona esta borracha, habrá movimientos aleatorios del chorro de orina. Su vista será desde arriba, simulando los ojos de la persona, es decir, lo que vería el usuario si realmente estuviera en esta situación (figura 1). El chorro de orina se comportará como un proyectil al caer, es decir, un movimiento parabólico.

Figura 1

* Carretera

Este nivel consiste en conducir un vehículo evitando chocar con algunos obstáculos, sean transeúntes u otros vehículos por el mayor tiempo posible, el usuario tendrá cierto porcentaje de vida y una vez se le acabe, el nivel terminara. Su vista será aérea, tipo ***Grand theft auto*** *2* (figura 2). Los modelos físicos que estarán presentes en este nivel serán movimiento rectilíneo en el movimiento de los vehículos y de los transeúntes intentando cruzar la calle, y choques elásticos e inelásticos en caso tal que el usuario choque con alguno de los obstáculos. De igual forma, sin olvidar el factor de aleatoriedad en el movimiento, debido al estado de embriaguez del personaje.

Figura 2

* Pelea en el bar

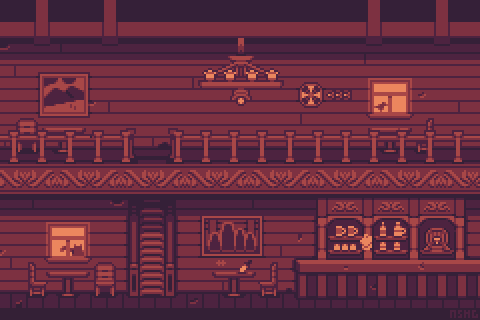
Este nivel está ambientado en un bar donde el jugador deberá coger botellas con más alcohol que caerán desde arriba evitando las botellas vacías y otros borrachos que irán de lado a lado por todo el bar hasta llenar la barra de puntuación al máximo, de igual forma que en el anterior nivel, el jugador tendrá una barra de vida que si llegase a gastar se acabaría el nivel. Su vista será frontal, y el jugador podrá saltar en algunas plataformas ubicadas en el escenario para evitar los otros borrachos (figura 3). El modelo físico presente en este nivel será básicamente caída libre de las botellas.

Figura 3

* Esposa

Si el usuario supera los tres niveles del juego, se desbloqueará este nivel a manera de ***easter egg***donde el personaje al final regresa a su hogar y debe rogar perdón a su esposa que le está poniendo problema por llegar tarde y borracho. Básicamente será un nivel muy sencillo pero realizado a manera de sátira si el tiempo lo permite.

No tiene físicas y simplemente será presionar una tecla lo más rápido posible para que el personaje ruegue a su esposa.

**Requerimientos**

1. **Tiro al blanco**

Requerimientos

Inodoro, orina, cuerpo (brazos y piernas), escenario, modelos físicos (movimiento parabólico), controles: arriba, abajo, izquierda o derecha, movimiento aleatorio, variación del área del inodoro.

**CLASES PARA ESOS REQUERIMIENTOS:**

-----------------------------------------------------------------------------------------------------

Inodoro:

**Atributos**:

Int Tamañox, int tamañoy (Área), string Color

**Métodos**:

VariacionArea(QTimer t) //Reducir el area del inodoro a partir de cierto tiempo t

**Signal**:

Void Acierto() //Señal que identifica si hay colisión entre la orina y el inodoro

Slot:

Void AumentarPuntuacion() //Slot que aumenta puntuación si hay colisión

-----------------------------------------------------------------------------------------------------

Orina:

**Atributos**:

String Color

**Métodos**:

QRectF boundingRect() const; //Identificar el chorro de orina (Un cuadrado muy pequeño que sería el final del chorro)

-----------------------------------------------------------------------------------------------------

Escenario:

**Atributos**:

QGraphicsLineItem\* l1;

QGraphicsLineItem\* l2;

QGraphicsLineItem\* l3;

QGraphicsLineItem\* l4;

//Cuatro barras que representan las paredes del escenario

**Métodos**:

Void UbicarEsc() //Ubica el límite de la escena donde puede apuntar

-----------------------------------------------------------------------------------------------------

Mainwindow:

Método:

Void Controles (QKeyEvent \*ev) //Identifica las teclas presionadas y además incluye un ligero movimiento aleatorio

1. **Carretera**

Requerimientos: vehículo del usuario, otros vehículos (diferentes tipos utilizando herencia), peatones, escenario, modelos físicos (movimiento rectilíneo y choques entre los diferentes obstáculos sean carros o peatones), controles: acelerar, frenar, cambiar de carril a la izquierda o a la derecha, porcentaje de vida, gasolina, puntaje, movimiento aleatorio, cronometro.

**CLASES PARA ESOS REQUERIMIENTOS:**

-----------------------------------------------------------------------------------------------------

Objetos nivel 2: public Qgrapichitems

**Atributos**:

string Color, int Velocidadmax, int Tamañox, int tamañoy, int Velocidad

**Métodos**:

Void Acelerar(bool)

Void Frenar(bool)

//Los dos métodos reciben un booleano que dependiendo de su valor modifica la velocidad del vehículo

-----------------------------------------------------------------------------------------------------

Jugador nivel 2: public objetos nivel 2

**Atributos**:

Int vida // esta es la vida que tiene el jugador. Se gasta cuando hay choques y con el tiempo.

**Métodos**:

Void gasolinacontrol(bool) //si el bool es falso se le quita gasolina y si es verdadero se le agrega. (el booleano es falso por el tiempo, cambia a verdadero cuando obtiene un ítem de gasolina en la carretera. SI es falso hay que ver si hubo colisión con carros o peatones).

Void choque () //desde acá se controla una posible colisión y según eso modificamos el atributo gasolinaporcentaje

-----------------------------------------------------------------------------------------------------

obstaculos: public objetos nivel 2

**Método**:

Void movimiento() // se define el movimiento que van a tener los otros carros.

-----------------------------------------------------------------------------------------------------

Escenario nivel 2: public Escenario

**Métodos**:

Void moverescenario() //el escenario se debe de ir moviendo o cambiando para simular que el carro esta avanzando

-----------------------------------------------------------------------------------------------------

Gasolina:public QgraphicItem //se hereda para hacerlo un item

**Atributos**:

string color, int tamañox, tamañoy

**Signal**:

Void sumagasolina() //envía una señal si hay que sumarle a la gasolina

**Slot**:

Void modificagasolina()

**Metodos**:

QRectF boundingRect() const // cuadrado

void paint(QPainter \*painter,const QStyleOptionGraphicsItem \*option, QWidget \*widget); //lo que nos dejara pegarle una imagen PNG

-----------------------------------------------------------------------------------------------------

Mainwindow

**Método:**

Void Controles (QKeyEvent \*ev) //Identifica las teclas presionadas y además incluye

**Slot**:

Void animar() // movimiento del escenario con todos los ítems y objetos

-----------------------------------------------------------------------------------------------------

1. **Pelea en el bar**

Requerimientos: personaje principal, porcentaje de vida, puntaje, enemigos (otros borrachos con diferentes características, por ejemplo, complexión física), botellas llenas o vacías, plataformas, cronometro

**CLASES PARA ESOS REQUERIMIENTOS:**

**-----------------------------------------------------------------------------------------------------**

enemigos:public QgraphicItem

**Atributos**:

Int tamañox, int tamañoy, int velocidad

**Métodos**:

Movimientogente() //van a aparecer de la parte lateral del escenario y van corriendo de un extremo a otro

-----------------------------------------------------------------------------------------------------

Personaje jugador nivel 3:public Jugador nivel 2

**Atributos**:

Int tamañox, int tamaño y, int velocidad

**Signal**:

Void colisionenemigos() //enviara una señal si entra en colisión con enemigos

Void colisión botellasllenas //envía una señal para añadir vida

Void colisionbotellavacia //envía una señal para bajar vida

**Slots**:

Void subevida()

Void bajavida()

Void nopasar() // slot que actúa cuando la señal de colisión enemigos se activa para no dejar pasar al jugador por donde está el enemigo.

**Métodos**:

Void movimiento () //movimiento de derecha e izquierda y salto

-----------------------------------------------------------------------------------------------------

Botella:public QgraphicItems

**Atributos**:

Int tamañox, int tamañoy, bool lleno

**Metodo**:

Void Movimiento()

// en el nivel 3 usaremos el mismo objeto Escenario del nivel 1

Decidimos quitar la clase peatones porque vimos que podríamos hacer algo muy general con vehículos ***npc***. También la clase escenario del nivel 2 y hacer una herencia de un solo objeto escenario.

Peatones: public Qgrapichitems

Atributos:

Int tamaño, int velocidad

Signal:

Void restagasolina() //emite señal si hay colisión con el peatón

Slot:

Void modificargasolina() //modifica la gasolina según la señal

Métodos:

Void movimientoPeatones() //se controla el movimiento del peatón.

Escenario:

Atributos:

QGraphicsLineItem\* l1;

QGraphicsLineItem\* l2;

QGraphicsLineItem\* l3;

QGraphicsLineItem\* l4;

//Cuatro barras que representan las paredes del escenario

Métodos:

Void UbicarEscnivel2() //Ubica el límite de la escena donde puede apuntar

Void moverescenario () //el escenario se debe de ir moviendo o cambiando para simular que el carro está avanzando

En el objeto Personaje nivel 3 decidimos quitar el método salto() porque vimos que en movimiento se podía modelar también.