



UNIVERSIDAD NACIONAL DEL LITORAL
FACULTAD DE INGENIERÍA Y CIENCIAS HÍDRICAS

Tecnicatura en diseño y programación de videojuegos

Modelos y Algoritmos para Videojuegos II

Unidad 4 Mouse picking

Docente

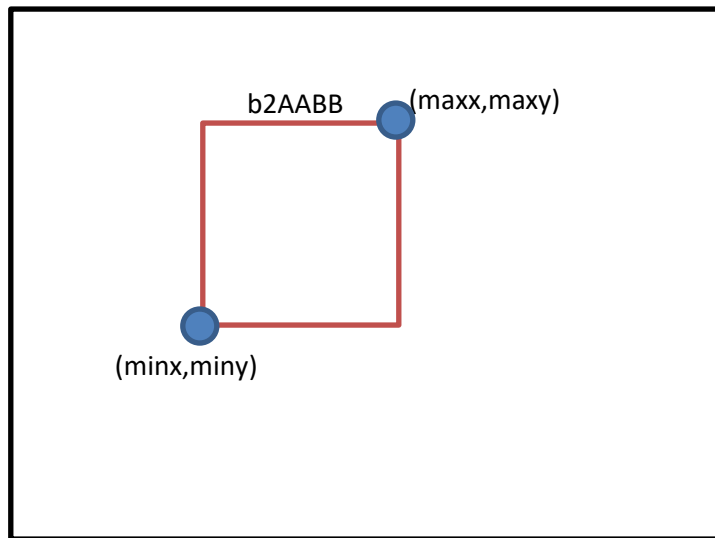
Lucas Sebastián Villa

Mouse picking

En este breve apartado vamos a ver cómo podemos hacer cuándo necesitamos encontrar objetos que se encuentren en una región determinada de la pantalla. En primer lugar, debemos definir cuál será la región de pantalla sobre la cual queremos buscar objetos. Para ello declaramos un objeto de tipo **b2AABB**. Este objeto define un rectángulo mediante su vértice inferior izquierdo y su vértice superior derecho. Por ello debemos asignarle los puntos de dicho rectángulo en coordenadas globales:

```
b2AABB box;
box.lowerBound= b2Vec2(minx,miny);
box.upperBound= b2Vec2(maxx,maxy);
```

Cómo se muestra en la figura



En dicha figura el sistema coordenado es el de Box2D no el de SFML. Deben tener en cuenta las diferencias que pueden surgir a partir de ambos sistemas coordenados.

Para realizar la consulta espacial, utilizaremos el método **b2World::QueryAABB(...)**. Dicho método recibe como argumentos el rectángulo sobre el cual verificar colisiones y una clase la cual recibirá las notificaciones de cada colisión que detecte. La clase que funcionará como receptora de las notificaciones de colisiones debe heredar de la clase **b2QueryCallback** e implementar el método heredado

```
bool ReportFixture(b2Fixture* fixture)
```

Dicho método será invocado por cada fixture que se encuentre en colisión con el rectángulo definido anteriormente. Por ejemplo, podríamos definir la clase de la siguiente manera:

```

class QueryDisparos: public b2QueryCallback {
public:
//Este método se llamará una vez por cada fixture que
//esté en colisión con el rectángulo
bool ReportFixture(b2Fixture* fixture){
    ...
    ...
    return true;
}
};

```

Y luego para realizar la consulta debemos crear un objeto del tipo **QueryDisparos** y pasárselo como argumento al método **QueryAABB** del objeto **b2World**. Dicho método recibe como primer argumento el objeto que utilizará para reportar las colisiones y como segundo argumento el rectángulo sobre el cual deberá realizar la prueba de intersección, por ejemplo, podríamos hacerlo de la siguiente manera:

```

//Creamos una clase que contenga el callback
//que será llamado una vez por cada objeto
//que colisiona con box.
QueryDisparos query;

//le seteamos la propiedades que deseemos al objeto
query.algo=...
query.algo2=...

//realizamos la consulta
phyWorld->QueryAABB(&query, box);

```

Al realizar la consulta por cada colisión se invocará el método **ReportFixture** del objeto **QueryDisparos**. En dicho método podremos realizar la tarea que deseemos sobre esos objetos.