**Cuestionario clase#5**

1. Los 32 de width y height significa que tiene 32 unidades de pixeles de ancho y 32 de largo. Pixel es la menor unidad homogénea en color que forma parte de una imagen digital.
2. Imagen 32x32, *Bounding Box:* Left: 3, Right: 28, Top:1, Bottom: 30
3. La máscara de colisión es el área que GameMaker Studio utiliza para calcular cuando dos instancias están en colisión o no y desencadenar un evento de colisión.
4. El origen del Sprite indica el lugar donde se centra la instancia del objeto que lleva esa Sprite en una celdas del Room, el origen se le puede asignar en toda el área del sprite utilizando las coordenadas X y Y. Tambien sirve para saber cómo rota el Sprite al hacer colición con un objeto sólido o al moverse.
5. Si un objeto sólido es True se genera un evento especial de colisión con lo que la instancia se vuelve automáticamente a la posición que estaba en en el paso anterior a la colisión.
6. Las instancias que saquemos de ese objeto se quedarán inmóviles, además para que el personaje no las traspase se coloca un evento de colisión en el objeto personaje y no en el objeto Wall, por lo que no se necesitan eventos para el objeto Wall..
7. Si no se le agrega al Obj\_goal la acción de pasar de Room el nivel nunca terminará, si no se le coloca la acción de reiniciar si hay un siguiente nivel, aparecerá un error que no le permitirá al jugador seguir jugando ni le determina si ya terminó el juego o no.
8. La Speed quiere decir qué tanto se desplaza una instancia con cada paso, las unidades de desplazamiento son pixeles.
9. 32 por 32
10. Porque si no se mantiene alineado los objetos y los personajes se chocarían y superpondrían
11. Es inusual y se bugea mucho.
12. Porque la alineación de personajes y grid evita que esto suceda.
13. Step es el paso a seguir si se cumplen ciertos requisitos.
14. Destruye la instancia en que se encuentra el personaje.
15. Se siguen ejecutando las acciones, puesto que no se elimina inmediatamente la instancia.
16. Que ese objeto comparte códigos y acciones con el objeto relacionado.
17. Se compartirían la programación, pero se les cambiaría el sprite a cada objeto.
18. Cuando se le añade algo a la programación del “hijo” estas acciones extras puede sobreescribir las acciones previas del “parent”.
19. Es un objeto que pone condiciones a las acciones en el juego, y de esa manera controla lo que sucede de diversas formas.
20. inicialmente un objeto sin acciones pero solo sprite se ve en el mapa pero no hace nada, al ponerle la acción de draw no sucede nada, ya que si el draw event no tiene una acción dentro de esta se toma como si no existiera este event, por lo que pusimos una acción dentro del event que era draw color y cuando se comienza el juego el objeto desaparece.
21. depth es la profundidad del objeto si es positiva el depth es más abajo si es negativa el objeto sobresale por encima.
22. ya el monstruo no pasa por encima de la bomba si no que pasa por debajo de ella.
23. significa que la acción que sucederá se la aplicara a …, en este caso escogemos que suceda a la bomba.
24. significa que haga la acción como en este caso fue una explosión de gif es que se performe el objeto.
25. significa que las instancias alrededor de la bomba se destruyen, por lo que relativo se toma como que exploten relativamente a la posición de donde se coloque la bomba.
26. Se aplica esta configuración para que el objeto aplique velocidad al other. El other es en este caso el personaje y lo que se hace es verificar si pa donde se mueve hay un objeto solido si si lo hay no lo mueve pero si no el other (personaje) empuja el objeto.