# Worksheet 3: Collision

|  |
| --- |
| **Learning Intentions**: We are learning to be able to … |
| Create and understand collision events in GM |
| *Why are we learning this?* |
| Collision formulates the basis of interaction for all objects in games. |
| **Success Criteria**: I will be successful if I can … |
| * Explain the types of collision * Diagnose and fix collision events that are erroneous * Implement collision events that utilise low CPU usage |

## Instructions:

|  |
| --- |
| Acquire and integrate knowledge: |
| 1. Use previous GM file but remove all events on all objects, put ball somewhere in room and leave wall around border of room:        1. Make **objBrick** “Solid”:      1. On **objBall**:   **Create Event:**    **Collision Event** *with objBrick*:      Play and test.   1. Does anything seem to change if I choose *not precisely* bounce in the above example?   Click here to enter text.   1. What do you think is the difference between random() and random\_range()? Explain how these functions affect the **speed** and **direction** of the object in the above example.   Click here to enter text. |
| Extend and refine knowledge: |
| 1. Add a *Key Press <Space>* ***Event*** + *Create Instance* ***Action*** to the object ball:      1. Why do you think we have ranged **64 – 576** and not **0 – 640** which is the full room width?   Click here to enter text.   1. Add another Collision Event to the ball, this time with itself:      1. Set Bounce action for Collision Event with itself:     Test and play.  Aside from the balls spawning on top of each other, you will find that the **game slows down a lot** – which can be seen visually by the *choppiness of the animation*.   1. If I change *precisely* to *not precisely* in the previous example, does it still happen as bad? What is the trade off?   Click here to enter text.   1. “Precise” or “per pixel” collision checking is expensive but accurate… so no collision will be detected here yet (which is correct):     This is using a “bounding box” on the aqua ball and is a lot cheaper on the CPU, but not as accurate (this should be a miss, but is counted as a hit):    On the sprite you can turn off Precise collision checking yourself, and Modify the bounding box Mask if you like:      Most games have “**hitbox**” areas, as per pixel collision is too slow… |
| Use knowledge meaningfully: |
| 1. Investigate what is meant by the Sprite origin. Good developers will always **Center** their origins. Explain what this does:     Click here to enter text.   1. Practise importing a sprite sheet. Use the **4\_web\_based/gamemaker81/sprites.html** resource to see how. |