G54GAM – Lab Session 3

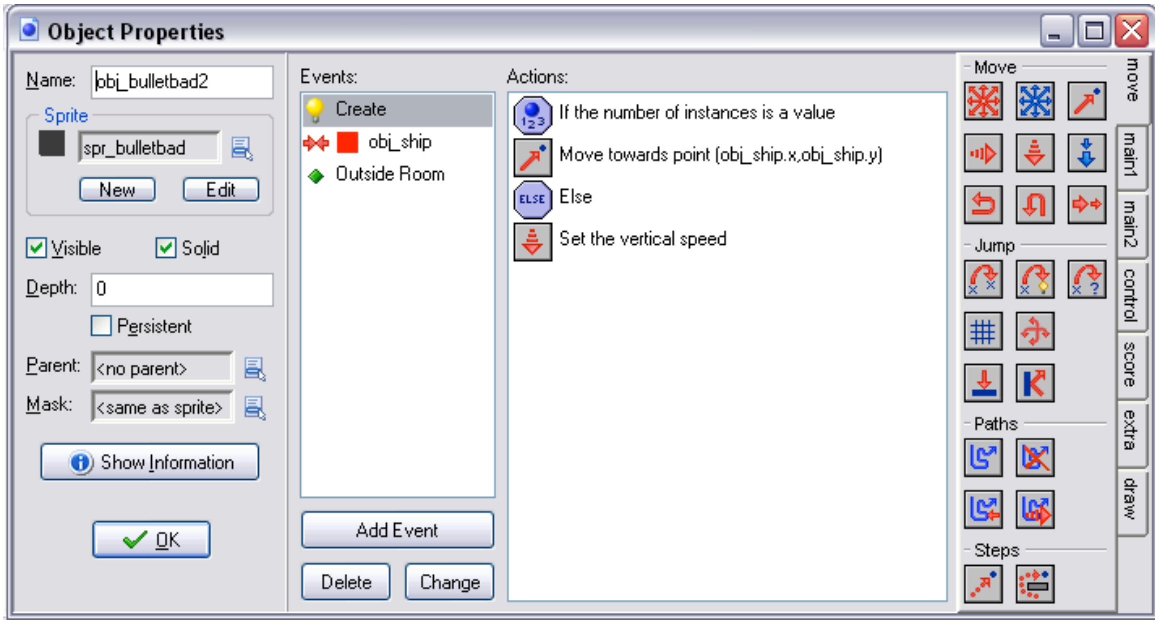
Scrolling Shoot-Em-Up - Extending the Game

Last week we created the basic mechanics or a playable shoot-em-up game. This week the aim is to extend it in a structured manner to make it richer, more challenging and to give it longevity, and hopefully meaningful play.

Better Enemies

Wave after wave of the same enemy isn't that interesting, so create a variety of enemies and place them so that the game gets harder as it goes along

- Create an object hierarchy of enemies with different sprites
- Have different enemy objects move at different speeds
- Create an enemy that fires bullets at you. You'll need to create an enemy bullet object that flies down the screen towards the player. Use the **Test Chance** action with a 1 in 30 chance per step that the enemy instantiates a new bullet object. Handle the collision between the main character and the enemy bullet to restart the game
- Create an enemy that fires targeted bullets. This is trickier, as you need to check if there's a character object to fire at using **Test Instance Count** (otherwise you're risking an exception) and then use the **Move Towards** action to aim at the player.



Name: obj_bulletbad2

Sprite: spr_bulletbad

☑ Visible ☑ Solid

Depth: 0

□ Persistent

Parent: <no parent>

Mask: <same as sprite>

Events:
Create
obj_ship
Outside Room

Actions:
If the number of instances is a value
Move towards point (obj_ship.x,obj_ship.y)
Else
Set the vertical speed

- Create a final boss enemy that appears at the top of the screen, then stops and remains in front of the player. Give it a health variable that is reduced every time a bullet hits it, and destroys the instance when the health drops below 0.

Lives, Health and Score

It seems a bit mean to restart the game whenever the player hits an enemy or an enemy bullet. Conveniently Game Maker has built-in support to keep track of **Score**, **Lives** and **Health,** so we can have the player's health reduce when they hit something, lose a life when they run out of health, and end the game when they run out of lives.

- Create an invisible controller object and place it in the room
- In the **Create** event, initialize the score, number of lives and health using the **Set Score, Set Health** and **Set Lives** actions. Also add a **Score Caption** action to show these values in the window's title bar.
- Have the controller object handle the **No More Health** event (in the **Other** event menu) to reduce the life count by one
- Handle the **No More Lives** event to show the scoreboard, using the **Show Highscore** action and restarting the game
- Return to the various **Collision** events between our objects. Instead of restarting the game, use **Set Score** to increase the score and **Set Health** to decrease the health a little depending on what the player has collided with.
- When the boss enemy has been destroyed, display the score board and restart the game.

Test your game with different values for the health decrements, and try and tune the game so it's hard, but not impossible, to reach the end and defeat the boss.

Coordination and Pattern Recognition Challenges

The movement of enemies as they fly towards the main character provides an opportunity to set coordination and pattern recognition challenges for the player – the player needs to avoid and shoot the enemies, or learn the patterns of how they move in order to shoot them to gain the highest score.

- Create path resources for enemies to move along
- Create a time line that spawns a wave of enemies that move along the same path
    - Use a time line and controller to create instances of a particular enemy
    - Destroy the enemies once they reach the end of the path using the relevant event
- Create a boss that enters and then endlessly circles the same path until the player manages to destroy them
    - Make the boss object repeat the path when it reaches the end
    - Use a local variable to define the boss's health, and destroy it when this runs out
- Create a palette of enemies that move and act in different ways

    - Flying straight down the screen
    - Flying towards the player
    - Flying along a path
    - A boss that flies along a path that requires multiple hits to defeat
- Create a challenging level that starts off with enemies that are easy to defeat, but that gets harder, culminating with a boss enemy

Think about:

- How the challenges might change as the game progresses, and the player's skill increases

Exploration and Progression Challenges

Challenge the player to work for their freedom to explore the game environment by creating exploration challenges in the form of multiple rooms, or levels, that the player has to progress through, and that provide structure to the game.

- Create a number of rooms with different backgrounds and a variety of islands
- Use the next room and different room actions to move between rooms
- Each room should have a different controller object and associated timelines to create a variety of obstacles – waves of enemies and bosses

Use multiple rooms to frame the game:

- Create a title screen room with a clickable object that starts the game
- On completion of a room, move onto the next room
- When the player dies, move them to a game over room before returning to the title screen
- When the player completes all of the rooms, reward them with a winning room

Think about:

- How do boss enemies present obstacles to the exploration challenge (Hint – the player must defeat the boss before they can start the next level/room)
- How can the player use their intrinsic knowledge of how previous enemies have moved to help them overcome the current obstacles?

Resource and Strategy Challenges

Currently our game offers the most basic of economic/resource challenges – the player has one life and infinite bullets, and if they use their one life then they have to start from the beginning. Challenge the player to make strategic decisions on how they use their resources in order to make their way through the levels:

- Use the built-in functions to create a basic economy of health, life and score for your game
- Extend collision and destruction events to increment health as appropriate
- Create a controller that handles the No More Lives and No More Health events
  - Decrement the life count when all health is lost
  - Increase the health when a life is lost
  - Display these variables to the user using the custom Draw event and associated Draw… actions

Consider the economic challenge of maintaining life and health, which are currently the main currency the player has within the game:

- How does the player consume this currency (Hint – they get hit by bullets, they run into enemies – they lose health)
- How does the player increase or produce this currency (Hint – they might collect items dropped at random by destroyed enemies, such as health and extra lives)
  - Create bonus objects that are spawned at random when an enemy is hit
  - When the main character collides with these objects, their health and lives are increased

What other economic resources might exist in the game?

- Power-ups that change how the main character moves or acts
  - Change the rate of fire
    - How long does this last for (Hint – use Alarms)?
    - How does the amount of damage change/how does it affect interaction with enemies?
  - Change the rate of movement
    - How might you change the speed of movement of the player?
  - Evil power-ups
    - Invert how the player controls their character
    - Decrease rather than increase the number of lives/health of the player

Think about:

- How use (and failure to use) the resources in your game prevent or enable the player in completing the game or level
- Are your resources utilitarian, scarce and timely?

We will cover these different types of challenge in detail in the next lecture!