# Game Off (2021)

Theme: **BUG**

Deadline: **1 December**

Requirements:

* Mostly new code/art/ideas for jam
* Put whole project on GitHub (open source)

Soft requirements:

* Single player mode available => most people won’t be able to test with multiple
* Web build available => just way easier and platform-independent

## General idea

**Everyone is a *spider* stuck on a (spider)web.**

This means you can only move over the current lines in the web. Obviously, you can *change* this web by shooting new lines or breaking old ones.

Is it **competitive** or **cooperative?**

* If competitive, it might be hard to create a solo player variant.
* If cooperative, it might be hard to find a good common goal.
* It’s also possible to make it a puzzle game … but do I really want that? (Could also try both … )

## Main Rules

Everything in the game is an **entity** (even the players) with a certain number of **points.**

* Entities can either move on the web (following the edges) or fly freely.
* If you encounter someone with *fewer points than you*, you eat them (and gain their point total).
* Species do not eat their own kind.
* Jumping across the web costs points.

Some entities have a **specialty**. This is some behavior unique to them. If that’s the case, *eating* them will always transfer this specialty to you (for a limited time).

All of this is altered, of course, in certain situations (such as entering a certain silk type or eating a creature with a special power).

## Movement

You can point in all directions. The system will find the line closest to your chosen angle and move along that.

If you press jump, you jump in the direction you aim, landing on the first line within reach.

* This creates a new line
* If there is no other side, or it’s too far away, you will not jump.
* (Alternative for hardcore players: you will fall off the web and die?)

The map also contains many “fixed points”. These can be used to attach lines, but are also a special location (such as a safe resting place, the default spawn for a predator, etcetera)

## ideas

**IDEA:** A **home base**. This is just a point on the web. Visiting this drains your points. (Either all of them, or X at a time.) This means they are *safe* there … but also makes you more vulnerable for a moment.

**IDEA:** A *water* map, where the *points* that connect the lines drift and float with the water. (And landing somewhere sends a shockwave around you.)

**IDEA:** A wind/dust map that can blow you off the web at certain points, causing you to react quickly or die?

**IDEA:** To “catch (certain) bugs” you simply need to trap them in the spiderweb. They will fly freely, but if they encounter a line of the web, they turn around. They have limited stamina, so if you’re able to trap them for X time, they become tired and will run into your web.

Or this only works for specific types of silk. Or only those owned by certain players.

**IDEA:** In general, play more with the unique *movement* of this game.

* A way to *curve* edges
* Or *slingshot* across the web.
* Or destroy/blast away all *points* in a certain radius. (Or attract them.)

**IDEA:** A predator that simply appears in-between edges. (Calculate polygons, then place them in the center of those polygons). From time to time, it shoots its tongue/claws to an edge around it.

**IDEA:** The title “Windowsilk” comes from the first arena, where you’re building the web on top of a window. Regularly, bugs just *drop* against the window (like a bird hitting a window it can’t see) for you to scoop up.

**IDEA:** The players can create their own bugs as well. (Offspring?)

**IDEA:** Levels might have fixed “spawn points” for bugs. (Like a home base.) They can destroy this, but doing so too soon isn’t smart?

**IDEA:** If players get big enough, they might even be able to *eat other players*.

**IDEA:** The “fungus” / “virus” idea => it starts somewhere on the web, and then just grows and grows, unless you’re able to stop it. (Cut off the edge, or use some powerup against it.)

# Collision Layers

These are the collision layers used by physics in the game

1. Spider web (edges and bounds)
2. Players
3. Collectibles
4. Points (they can be *moved* by things such as wind or water, and are thus KinematicBodies that collide with each other)

# Ownership rules

When you create a new line, it becomes *yours*.

How does ownership work? Some ideas:

* You *can* travel over other colors, but it costs 1 silk each time.
* The owner wears off after a while.
* Lines only become yours if *the jump is big enough*
* Lines only become yours *if a certain powerup is active*.

(Team members can also travel over your lines, of course.)

# Entities

## Web

### Larva

**Points:** 1

**Move:** doesn’t

**Silk:** none

**Specialty:** can always be eaten

This is the “fail-safe” of the game. If you have 0 or 1 points, this is the only thing you can eat. (Any other entity will have more points *or* move around faster than you.)

### Tiny spider

**Points:** 1

**Move:** slow

**Silk:** regular (?)

**Specialty**: none

This is the backbone of the game. They are quite easy to catch and appear often, but do nothing special and will not be enough in the long run.

### Flea

**Points:** 2

**Move:** fast, but pauses longer on points/needs breathers (?)

**Silk:** speedy (*you move faster*)

**Specialty:** none

The reason they create speedy silk is *also* because it helps them flee. (If they exit an edge and see a predator, they backtrack over the edge they just painted, which is *fast*.)

And yes, “flea” sounds like “flee”, hopefully it helps people remember what they do :p

(This has a twin in the *flying* bugs that *slows you down*.)

### Silverfish

**Point:** 3

**Move:** extremely quickly, non-stop

**Silk:** slippery (*you move like you’re on ice*)

**Specialty**: none

The silverfish is named like that *because* it’s slippery. (It’s not an actual fish.) It moves quickly and gets everywhere, but you can never catch it.

### Grasshoppers

**Points:** 4

**Move:** only *jumps* (without creating new lines, of course)

**Silk:** trampoline (*jumping is free*)

**Specialty:** when eaten, jumping is free for X seconds (Alternatively: you can *only* jump, not move regularly.)

### Locusts

**Points:** 1

**Move:** *can* jump, doesn’t always do it (then just shuffles around)

**Silk:** doubler (*anything eaten here counts double*)

**Specialty:** regularly *multiply* => a new locust is added next to them

Locust plagues are a thing. Their strength is in *numbers*. Whenever a locust appears, you want to be on top of them *immediately*, before they become annoying.

### Crickets

**Points:** 5 (*friendly*)

**Move:** nothing special

**Silk:** noisemaker (*instead of jumping, you make noise that blasts away entities around you*)

**Specialty:** once in a while they make noise, blasting away any threats near them.

Crickets are known for their chirps, so this felt logical. They also have basically no way of threatening anything (very soft mouth, rarely bite) and are lightweight, so *friendly* seems fitting.

Fun fact: they chirp faster if it’s hotter outside. See no clear way to include this in the game (except for a special, hot level?), but still fun.

### Cockroach

**Points:** 4

**Move:** fast, fleeing

**Silk:** none

**Specialty:** cannibals; eat their own species, even chase them

Because they eat their own species, and are rather big to begin with, they can quickly grow in size.

### Beetles

**Points:** 6

**Move:** slow, might start flying from time to time

**Silk:** shield (*protects you against being eaten*)

**Specialty:** when eaten, you gain a shield for some time

I chose beetles for this as they are known to have a really hard shell that looks *somewhat* like a shield.

### Flightless Fruit Fly

**Points:** 1

**Move:** shuffle, flee, medium speed

**Silk:** regular (so it basically *erases* any existing terrain types)

**Specialty:** erase

This bug is very important to prevent the web from getting too complex (with *every edge* having its own terrain).

I chose the fruit fly because there is a flightless and regular variant, both eaten by spiders. (The “erase” functionality is so important that I want an exact copy in the “flying bugs” department.)

## Flying

### Regular Fruit Fly

Identical to the flightless fruit fly. But this one flies.

### Wasp (regular)

**Points:** 0

**Move:** erratic

**Silk:** worthless (*anything eaten here is worth nothing*)

**Specialty:** (the fact that it’s worth nothing)

I hate wasps. They are worthless to me.

However, you still want to eliminate them. If you leave them roaming too long, *everything* becomes worthless.

### ??

**Points:** 2

**Move:** slow, but speeds up when it sees prey

**Silk:** slowy (*you move slower*)

**Specialty:** none

The idea is that they slow you down with their silk. So that when you come near, you’re a much easier target for them.

### ??

**Points:** 4

**Move:** just hovers in empty space

**Silk:** sticky (*you cannot jump away*)

**Specialty:** none

The idea is similar to the “slowdown”: it hovers near edges, and as bugs cannot jump away, it’s easy for them to catch you on their silk.

**IDEA:** Something that only moves over unowned silk.

**IDEA:** Something that never backtracks (where it’s already been). Or, conversely, something that always backtracks (and just moves up/down)

## What is food for spiders?

**Web**

* ~~Crickets~~
* ~~Grasshoppers~~
* ~~Roaches~~
* Beetles
* Earwigs
* ~~Fleas~~
* Ants
* ~~Locusts~~
* ~~Silverfish => not an actual fish, include those as well?~~
* Mealworms
* Small Caterpillars
* Flightless Fruit Flies

**Flying**

* Flies
* Butterflies
* Mosquitos
* Moths
* Bees
* ~~Wasp~~
* Hornets
* Gnats
* Fruit Flies

## What are predators for spiders?

* Birds (Great Tits)
* Lizards (Geckos, Chameleons)
* Frogs
* Toads
* Tarantula Hawks (insect, not a bird)
* Spider wasps
* Monkeys
* Centipedes
* Scorpions
* Other spiders. (Mainly female spiders eat the male, if it’s smaller than them.)
* Fish
* Bats
* Shrews

# Silk types

Some ideas for silk types.

## Movement

* **Regular (tiny spider)**
* **Speedup (flea) =>** you move faster over it
* **Slowdown** (??) => you move slower over it
* **Slippery (silverfish)** => you keep sliding (even when you stop moving) and have trouble turning around (quickly)

## Jumping

* **Trampoline (grasshopper)** => jumping is free
* **Sticky** (??) => jumping is forbidden
* **?? Cheap** => jumping is much cheaper ( = costs less silk)
* **?? Expensive** => jumping is more expensive ( = costs more silk)

## Web

* **Aggressor** => if you try to jump from this silk, you destroy the other side instead
* **Strong** => cannot be broken
* **TO DO: Fragile** => once an entity leaves it, it breaks
* **TO DO: Timebomb** => after X steps on it, or entities spending Y time on it, it breaks
  + This might also be when you *enter* it, creating this sense of: “has this silk already been worn out or not??”
* **Unsteady/Featherlight** => the more weight you put on this strand ( = more entities on there), the more it *moves*
  + It moves the outward points towards the center.
* **One-way traffic =>** the icon points a certain direction; that’s the only way you’re allowed to walk

## Collecting

* **Worthless (wasp regular)** => collecting something here *does nothing*. (Just removes it. Takes it away from anyone else.)
* **Doubler** **(locust)** => collecting something here gives you *double* its value.
* **Shield (beetle)** => you cannot be eaten

## Miscellaneous

* **TO DO: Noisemaker (cricket)** => instead of jumping, you make noise that blasts away entities around you

# To do

**Move behaviors:**

* **Jump** => add Timer underneath Movement
  + When it triggers, try random directions until we find an existing edge
  + Now ask Jumper to tween to our final point on that edge
  + For **flying** animals, just pick a random direction (not out of bounds) and tween to quite some distance ahead
* **Shuffle** => create Timer underneath Movement, which is rather quick and calls “pick\_opposite\_vec”
* **Stamina** => any time we moved, send a signal that subtracts the distance from total => if below 0, disable moving and start timer for re-enabling (also underneath Movement)

**General things needed:**

* Ability to pick different type/number of legs. As well as remove them entirely for flying. (Although some flying things actually spread out their legs, like beetles. Allow that?)

**Specialties:**

* Create *another* module for this. (So collector really only *collects*, not *executes consistent powerups*.)
* Trampoline
* Doubler
* Noisemaker
* Shield
* Erase

**Silk Types**

* Slippery => doesn’t really work. It needs to keep your *momentum*, which means I’d need to use a module\_update in the mover and use *that* for moving.
* Noisemaker

**Effects:**

* When jumping, make the new line appear *gradually* (out of our butt :p)
* When removing lines, do the opposite and make them disappear gradually
* Dying animation and feedback
* Silk change tween
* Feedback

## Optional

**Spider animation (Improved!):**

* <https://www.youtube.com/watch?v=e6Gjhr1IP6w>
  + Yes, interpolated movement
  + *Start* the legs in a zigzag movement
  + Only move a leg if the others (“supporting ones”) are grounded.
* <https://www.youtube.com/watch?v=LNidsMesxSE>
  + GDC talk about it, might be interesting in any case
* Do a *intersection check* to find any surfaces near that area, then reset to a point on them?
* *Interpolate* the resetting (instead of making it instant)?
* How to ensure legs go in alternating patterns?
  + Maybe *queue* resets. Each frame, check the queued resets. We only allow it to continue, if the surrounding legs are in the right position.
  + Example: a leg wants to reset. Then the legs before and after it should be reasonably far forward (low distance). The leg on the other side should be reasonably far forward as well.

**Web** (shooting/management/etc.)

* Now it doesn’t allow jumps that are too short … is that even a good idea?

**Manual starting levels (instead of randomized)**

Create a scene for each.

* In the editor, create Line2D nodes.
* Upon load, convert their end points to actual points. (Snap to existing ones, so I can be imprecise.)
* And then connect the edges.