# 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.
* The more points you have, the *bigger* you are and the *slower* you move.
* If you encounter someone with *fewer points than you*, you eat them. (Their points are added to your total.)
* Species do not eat their own kind.
  + Exception: players can eat other players, but only if the point difference is big enough (>5)
* Jumping across the web costs points.
  + This creates a new line in the web which is *yours.* Opponents cannot enter and flying bugs will get stuck in that line.

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.)

# Procedural Animation

These elements are procedurally animated: **legs, antenna, wings.**

This means no animations are manually created/predefined, but it calculates the right lines/positions/rotation *at runtime*. For this to work (well), I do still need to specify some things per individual bug.

In general, these properties exist:

* Color => the color of the leg
* Scale\_thickness => how thick the lines should be (scaled against default)

## Legs

Create nodes for the *starting point* of each leg. Name them “L1,L2,L3…” and “R1,R2,R3…” for the left and right legs.

Within each, add another node which is the *end point* of the leg. Call this “Offset”.

The legs use the most basic model for procedural animation:

* Each leg has an “ideal” position or “resting” position. This is what the nodes above define.
* When we move, the endpoint of the leg just stays on the ground and doesn’t move with us.
* When the leg is *too far from the ideal position*, it snaps back to the front.

Some improvements were made, and more can still be made, but this is the idea.

## Antennae

Create nodes for starting points, named **L** and **R.** Within each, create another node for the end point called **Offset.**

Optional: create a third node called **Mid** which acts as a control point, if you need to curve or bend the antenna.

The antennae use a (*heavily* simplified) physics simulation. The end point is attached to the start using an “elastic” or “rope” relationship, allowing it to swing back and forth and react to the bug movement, within limits of course.

## Wings

Wings are too complex to draw with lines (on the fly), so I use sprites. For each unique type of wing:

* Create a child named “L” with another child named “Sprite”.
* Change the “offset” on the Sprite so it’s centred around its pivot point. (The point where the wing would rotate.)
* Place “L” correctly on the bug.
* Set the sprite to the correct frame, obviously.

**Important:** at runtime, it automatically duplicates the wing and inverts it. This only works if you **don’t** change the Y-offset on the Sprite and its roughly in the center of its parent (L).

Two simple tweens will handle the “wing flapping” (during flight) and “wing collapsing” (when it lands), no matter what we put in.

Custom properties:

* min\_rot: the rotation of the wings in collapsed ( = landed/resting) state (default is 0.1 PI)
* max\_rot: the rotation of the wings in flight (default is 0.35 PI)
* show\_in\_front: show the wings in front of the body (of the bug) (default is false)
* collapse\_using\_scale: whether collapsing the wings should scale them down, instead of rotating them (example: butterfly)

# Soundtrack

As usual, I want to compose a proper soundtrack for the game *taking into account the theme and gameplay*.

The themes are: spider, spider web, bugs, catching/eating (and the tension that comes from fleeing/chasing)

The “spider” part was the most unique and concrete, so let’s start with that. Spiders have 8 legs (which is actually quite rare, as I’ve learned through making this game).

So let’s try to:

* ~~Compose with chords that consist of 8 notes~~ => Possible, but not really a sound I want for this game, quickly sounds too “bombastic”
* Create melodies from 8 notes => Easily done, but not that special.
* ~~Create chords with 8 notes, where each time~~ *~~pairs of the notes move~~* ~~(like a spider walking)~~ => sounds nice, but not for this type of game (more like a movie score during a sad moment)
* Create a melody with very rapid succession/alternating between notes (like a spider walking) => Great! Use it a lot. But not too much, as it’s a bit repetitive/grating.
* Play with the *meter:* start with 8/8 measures, dip down to 7/8, then 6/8, until we’re at the bottom (3/8? 2/8?)

On top of that, we can add distinct bug sounds that have some musical quality:

* The sound of crickets
* The buzzing of wings
* The shuffling you hear when a spider (or something else) crawls over something
* (Eating/munching sounds, although that’s probably not great, and would be a sound effect in the game itself anyway …)

**I don’t want to do another “solo-piano-focused” soundtrack.**

* Start with cricket sound => this sound has a repeating synth-like high note, and a low shuffling beat. (Only issue is that it’s not perfectly following the beat, so might need to quantize/recreate this myself?)
* 8/8:
  + Play the main melody on top.
  + Play the quick alternating notes
  + Play the full chords (pitch them to correct one)
* 7/8:
  + Play the quick melody on F (12 12 123)
* 6/8:
  + Play the “positive tune” repeated melody
  + Invent another melody
* 5/8:
  + Invent another melody?
  + (Listen to Mission Impossible theme to get a sense for the beat.)
  + (Even better: “When your mind’s made up” by Glen Hansard. => 123 123 12 12)
* 4/8:
  + Play the very quick D minor melody
  + Something else? (Move 8/8 melody to here?)
* 3/8:
  + Invent a 3-node melody
  + (Or just a waltz => check how that sounds and if I can use it)
* 2/8:
  + Invent a “sailor” or “march” melody. Basically slow down the other melodies, slower and slower (so it’s obvious the soundtrack is nearing its end)

Then repeat.

Lessons learned about music:

* 5/4 = 123 12
* 5/8 = 12 123
* 3/8 = **1**23 **1**23
* 6/8 = **1**23 456

# Home bases

Why are they in the game?

* Any numbers above 10 are just boring and break the game. It’s a basic case of “power creep”: once you’re too big, you can just eat anything, and you’re too powerful to stop for anyone.
* (Also, it’s hard to calculate with these bigger numbers. Keeping them small makes the game simpler and more accessible.)
* By capping your max at 9, you must constantly go back to base and deliver your points.
* Additionally, this restarts you from 0, creating tension again and new situations.
* Lastly, it provides a nice constant visual on your *target* and *how far you are to reaching it*, and encourages teamwork (as everyone in the team adds points to the same pool)

They do, however, need some extra work. These bases must *always be accessible* (without jump/powerup/points) and opponents should *not be able to camp them*.

To keep them accessible:

* When generating, the home base always gets at least X edges.
* Before an edge is removed, it checks if it’s attached to a home base with (fewer than) X edges. If so, disallow it.
* Similarly, edges attached to a home base can’t be painted.

To prevent camping, I simply make it a very bad strategy.

* Players with 0 points (which have just delivered/reset to their base) cannot be eaten by other players.
* In fact, anyone *near* their home base is invincible.
* There’s a high chance the silk around the home base is *owned* by players from that team, which means you can’t move over it *and*
* Because there are no special edges near a home base, you get no benefit from being there.

# Tutorial

How are we going to teach this game?

It’s a simple game with only a few rules. The bugs themselves are completely independent and can be taught/toggled one at a time.

**Show objective + main rule any time a round starts**. (This helps any newcomer get the gist and differentiates between different modes.)

* Objective? “Score X points (per player in a team)”
* How? “Eat creatures with less points than you.”

**If enabled (which is the default), show interactive tutorial next to players.**

* Move with <these keys>
* Jump with <this button>. Jumping costs points.

In the settings, players can choose which *mode, bugs* and *bonus rules* to use.

The bugs are split into “Web Bugs” and “Flying Bugs”. (Otherwise it’s too much on one screen; this is the most obvious categorization.)

Next to the selection grid is an image quickly explaining the different elements:

* Trail = bugs change the edge they are on
* Specialty = some special behavior. When you eat this bug, you temporarily get this behavior as well!

# Marketing

(*Marketing* just means: what text and images will appear on the page? A Game Jam game will not have a full marketing plan :p)

**Blurb:** 1-6 spiders compete to eat more bugs than the others, growing and changing their spider web as they go.

**Blurb:** **Windowsilk** is a party game for 1-6 players about eating … or being eaten. Run over the spider web to trap bugs, jump at the right moments to create new lines or catch flying bugs mid-air.

### Eight facts about the game

Here are 8 facts about this game.

* **Player count?** **1-6 players**.
* **Input?** Keyboard or controller, any combination you want.
* **Complexity?** Easy. Only two buttons and one major rule to understand.
* **Bugs?** 26 of them, all unique.
* **Arenas?** 5 cool places to build webs.
* **Why?** This game is for a Game Jam with the theme “Bug”.
* **Tools used?** Godot Engine (game software), Affinity Designer (graphics), Studio One (soundtrack)
* **Source?** Freely available on GitHub: <TO DO: Link here>

Although the game is playable online, we do recommend downloading it and running locally if you’re playing multiplayer. Allows for a bigger screen, better performance, and more stable support for input devices.

### How to play?

The rules are simple:

* Bugs can eat bugs with fewer points than them
* Walking leaves a trail on the web. Any bugs flying or walking into a player trail might get stuck, giving you a free meal.
* Jumping across the web costs points. (But it’s faster and extends the spider web with new lines.)
* Visit your home to deliver your points and restart from 0.
* The first team to have X points (depends on settings) at home wins.

This makes the game accessible and easy to learn, as it requires only two inputs: move and jump.

### Bugs

This game is all about **bugs.** There are no powerups, the bugs *are* the powerups. There are no other rules or systems to remember, the bugs do it all.

Every bug in the game has a unique *trail* they leave behind, which changes the properties of lines in the web. A Grasshopper makes jumping free, everywhere they go. A Fly allows players to grow wings and take off from the lines it paints.

When you eat a bug, you get this property as a powerup! Eating a Grasshopper makes jumping free. Eating a fly allows you to fly.

It’s that simple. But when more and more bugs are added, which might be bigger than you, or be unfriendly to begin with (looking at you, hornets), things get tense.

The game has **26 unique bugs**. Which, as you know by now, also means 26 unique line types and powerups.

Two games are never the same. And which bugs you use (or purposely *don’t* want to use) is completely up to you.

### About

Hi! I’m *Tiamo Pastoor* (otherwise known as *Pandaqi*), an indie developer who (almost) exclusively makes local multiplayer games. Try some of my other games as well:

* A Recipe for Disaster <TO DO: Links>
* Totems of Tag
* Carving Pumpkins & Dwarfing Dumplings

I also wrote a detailed devlog about the development of this game: <TO DO: Link here>

As usual, the starting idea was simple … then problems happened … and the final product is absolutely nothing like the original idea. Writing a system that allows natural movement *for so many different bugs/powerups*, both on and off a spider web, was a challenge. And when I thought it’d be fun to allow *moving* the points of web as you go, I silently cursed myself.

If you’re interested in any of the game’s mechanics, the source code is freely available here: <TO DO: Link> (As per requirement of the GitHub Game Off jam.)

If you have any **feedback**, let me know. Games made for jams can always improve immensely, especially one as ambitious as this.

There were several ideas I couldn’t implement (due to the time constraint). These might be added later. But by all other accounts, the game is completely finished.

# 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:** featherlight

**Specialty**: featherlight

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:** lowlife (*cannot enter if you have an active powerup*)

**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.)

### Aphid

**Points:** 2

**Move:** slow

**Silk:** fragile

**Specialty:** fragile (*anytime you exit an edge, it’s destroyed*)

Aphids are “Leaf fleas” or “Greenflies”, which means they are known for covering leaves and chewing on them relentlessly. That’s why it seemed fitting to make them turn silk fragile.

### Mealybug

**Points:** 2

**Move:**

**Silk:** sticky (*nobody can jump away*)

**Specialty:** sticky

Mealybugs are (oddly cute) white bugs that leave a sticky substance behind everywhere.

### Ants

**Points:** 5

**Move:** nothing special.

**Silk:** strong (*cannot be broken*)

**Specialty:** strong (*your current point/edge cannot be deleted*)

Most people know that ants are very strong and can carry stuff many times their own weight. Trying to use that knowledge to make this seem intuitive.

### Mealworm

**Points:** 2

**Move:** moves in short sprints (low stamina, high speed), moves like a worm

**Silk:** timebomb (edges are destroyed after one visit/several visits)

**Specialty:** timebomb => **how to copy to a specialty? Don’t see it …**

Couldn’t find a better place for this. Felt like either a *really small (“fragile”) bug* would fit, or one that’s actually huge and therefore breaks stuff.

### Small Caterpillar

**Points:** 3

**Move:** moves like a worm as well

**Silk:** gobbler (*you can eat anything, no matter the points*; might rename to *Hungry*)

**Specialty:** gobbler

Most people know the story of “The Very Hungry Caterpillar”. Piggybacking that.

### Earwigs

**Points:** 2

**Move:** chases, cannibal

**Silk:** aggressor

**Specialty:** aggressor

Earwigs have these huge pincers on the back of their body. (Many beetles have them, but I already used the beetle for something else). Looks aggressive.

**Although earwigs do have really cute babies**. Use that for something?

## Flying

### Regular Fruit Fly

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

### Fly

**Points:** 1

**Move:**

**Silk:** flight (*pressing “jump” makes you fly freely – when released, you snap to the closest edge* => if none nearby, you simply die)

**Specialty:** flight

Obviously, the plain old fly was the best candidate for making players fly.

### Wasp

**Points:** 0

**Move:** erratic/shuffly, doesn’t land

**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.

### Gnat

**Points:** 2

**Move:** slow, chases (**speeds up when it sees prey???)**

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

**Specialty:** none

These are extremely tiny and slender mosquitoes (usually mistaken for babies). They’re not exceptionally slow, but at least slower and calmer than the others.

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.

### Butterfly

**Points:** 5

**Move:**

**Silk:** attractor (*pressing “jump” attracts any nearby insects towards you*)

**Specialty:** attractor

Butterflies are often seen as the “beautiful” type of insect, attracted by flowers. That’s why attraction seemed to be a good fit.

### Bee

**Points:** 5

**Move:** not especially quick, but smoothly/in straight lines

**Silk:** time gainer (*for each X seconds you stay here, you gain a free point*)

**Specialty:** time gainer

Bees are good, cuddly, friendly. That’s why they give a very positive powerup.

### Moth

**Points:** 2

**Move:**

**Silk:** one-way (*you can only move across this silk in the direction indicated*)

**Specialty:** one-way => once grabbed, you cannot turn around until it wears off

Moths are usually on a one-way street to any nearby light source. I know, a vague reference, but the best I could do for this powerup.

### Hornet

**Points:** 9

**Move:**

**Silk:** poison

**Specialty:** poison

Huge kind of wasp. Aggressive-looking. Poisons anything it touches.

As with everything in this game, the poison also affects itself. That’s why it starts with so many points: it will go down over time.

### Mosquito

**Points:** 3

**Move:** chase

**Silk:** time loser

**Specialty:** time loser

Everybody hates mosquitoes, they’re just a waste of time and energy, they drain your blood. Hence the “time loser” specialty.

## 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~~
* ~~Aphids~~
* ~~Mealybugs~~

**Flying**

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

**Insects I missed (which are quite well-known):**

* Ladybug
* Dragonfly
* Centipede
* More other beetle types (that look completely different + have some wild colouring)

## 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** **(gnat)** => 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 (mealybug)** => jumping is forbidden
* **?? Cheap** => jumping is much cheaper ( = costs less silk)
* **?? Expensive** => jumping is more expensive ( = costs more silk)

## Web

* **Aggressor (earwig)** => if you try to jump from this silk, you destroy the other side instead
* **Strong (ant)** => cannot be broken
* **Fragile (aphid)** => once an entity leaves it, it breaks
* **Timebomb (mealworm)** => destroys itself once entities have walked over it for X seconds total
* **Featherlight (tiny spider)** => 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 (moth)** => the icon points a certain direction; that’s the only way you’re allowed to walk

## Collecting

* **Worthless (wasp)** => 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
* **Gobbler (caterpillar)** => you can eat *anything*
* **Time Gainer (bee)** =>For every X seconds you stay on this edge, you *get* a point.
* **Time Loser (mosquito)** => For every X seconds you stay on this edge, you *lose* a point.

## Aggression

* **Poison** **(Hornet):** instead of eating something, you *poison* it. This slowly drains their points and makes their movement slower/more erratic.

## Miscellaneous

* **Noisemaker (cricket)** => instead of jumping, you make noise that blasts away entities around you
* **Attractor (butterfly)** => something that *attracts* others (reverse of noisemaker)
* **Lowlife (cockroach)** => cannot enter edges with powerups

# Arenas

## Windowsill

A dusty window, basic arena. (The game is named after it!)

The fact that it’s a window would allow other insects to *splat* against it. Implement that?

## Water

The currents constantly move around the points.

All movement is “slidy”.

*Add fishes/water spiders? Some extra danger that randomly pops from the water and eats bugs/points?*

## Windstorm

Gusts of wind blow insects in different directions. (Like the noisemaker/attractor.)

These gusts can also obstruct view sometimes (with dust clouds).

## Forest

TO DO ??

## ??

TO DO ??

# Ideas

**IDEA:** More chase/flee types. (As it’s become quite a big part of the game.)

* **Naïve chase:** also chase if you don’t have enough points to eat something
* **Naïve flee:**also flee if someone isn’t even a threat
* **Conditional chase/flee:** only do it if the point difference is big enough, or the other is faster/slower than you, etcetera

**IDEA (“Bugging”):** Somehow, you can plant an *egg* in someone else’s home base, and steal 1 point every time someone delivers.

**Continuing on that idea:** maybe you can plant a bug *on* another player. So anytime they eat something, some % of that goes to you.

**IDEA:** Something more directly related to your *point total* => you need at least X points to enter, the more points you have the faster you move, etc.

**IDEA:** A “hidden” or “joker” type => you only know what you (randomly) get, when you enter/use it.

**IDEA:** Something with eggs or babies. If you’re big enough, you can lay an egg (which drains your points). The egg must be protected until it hatches. Hatch X eggs to win.

**IDEA:** Something with an *actual attack* against predators? (Like, if you try to eat it from the back, it will/might sting you. Or, once in a while, it just shoots something.)

The only problem is that this adds loads of complexity and exceptions. If I can find bugs with clear “attacks”, which still 100% follow the handful of rules in the game, it’d be great.

**IDEA:** Something that, once in a while, **blows up in size** (gets +5 or +10 points) and starts chasing everyone around them for a few seconds.

**IDEA:** The bumper => your body becomes solid, so that when you walk into another entity (without any eating occurring), you literally bump them away.

**Continuing on that idea** => the glue => bumping into a bug will stick it to you, so you can drag it to somewhere else where you *can* kill/eat it

**IDEA:** If you’re stuck, but the owner wears off … get unstuck again?

**IDEA:** Extend trails to *points*? => Only add the “trampoline” trait to certain points, so that jumping from them is free?

# To do

**Menu screens:**

* Make overlay settings menu work => include GlobalConfig, copy codes and scenes from Pumpkins
* Create “arenas” and “bugs” custom web
  + Give them their own “start” nodes (so players automatically position there)
  + And a clear exit location (same icon as Quit)
* Distribute all values in our config across the other points ( + show icon on the point itself)
  + Make icon dynamic: just listen to whatever we’re set to
  + Reserve space for the general “How do bugs work?” explanation
* Disable players logging in
* When walking over a point, you get a prompt to toggle it on/off

**Question:** what does *Poison* do as a silk type? (Right now, it makes everything you eat simply poisoned. Is that clear and good enough?)

**Tiny issue:** not enough space on triangle home base to hold more than 1 team member … (And probably other shapes as well.) => Once I can log in multiple players (in a single team), test this again

**Essential gameplay:** **Solo mode.** Do you lose if you die X times? Is there a timer?

**Visuals:**

* Home bases:
  + Add some flair around the edges to make it look like an actual cozy home. (Some leaves, some extra (tiny) spider web, some texture/gradient near the edges.)
* Add *shadows* underneath the web?
* Create some basic arenas, so I have an idea how the final game looks => especially the Window (sill) Arena.

**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
* Silk change tween => maybe a gradual color fade? (like, from one end of the line to the other, it changes)

**Before publishing:**

* Ensure quick-death is turned off
* Ensure quick-gameover is turned off
* Remove debug\_edge\_types from starting generation

## Plan Until Finish

I’ve planned to participate in another game jam (and in general have more stuff to do), so I’d like to finish this game before then. Here’s an agenda:

* **13 Nov:** Create interactive menu + picking settings
  + Build the mockup I created into a functional menu
    - Put the code for “register new player/deregister” (based on event) into the GlobalInput script
  + **Question:** change this so that *any* button jumps? (And the start/select buttons open the menu?)
  + **Question:** change this so you just “press once” to start jump, then “press again” to execute?
* **14 Nov:** Further essential polishing and balancing
  + Add final sound effects + particles
  + Soundtrack
  + Balance the properties of individual bugs + find best starting setup (and any other constraints needed)
  + Prettier home bases, lines, points, etc.
  + Different arenas and rules
* **15 Nov:**
  + Create itch.io page ( + already prepare portfolio and pandaqi one)
  + Create a logo (both header/wide and favicon)
  + Schedule a quick *playtest* for it.

## Particles

* When moving => some basic dust clouds from our butt. (Re-use the trick to make those show your *team*?)
* When eating => some lines shooting away from us
* When delivering to home base => maybe a more general “lose/gain points”
* When dying
* When jumping => a stronger, more “wind”-like variant of moving
* When someone gets stuck
* When a new line is created (or point)
* When an existing line is destroyed (or point)
* Specialties:
  + Poison => poison icons emitting from you
  + Noisemaker/Attractor => circles going outward or inward

## Sound effects

* Moving => leg shuffling, wing flapping/buzzing, jump/whoosh
* Eating => a munch sound
* A sloshing sound for worm movement
* Getting points/Delivering home => a positive tune
* Losing points/Dying => a negative tune
* Getting stuck => something like a bow that snaps? An elastic pulled back?
* Point/line creation and destruction => similar to the getting stuck sound.
  + I just need something with a ropey/elastic/silky sound
  + Both in a “snap/strengthen” sound and a “snap/break” version
* Specialties:
  + Noisemaker => a cricket noise
  + Attractor => an … attractive/angelic tune?
  + Poison => ??

## Optional

**Interesting stuff (Perlin Worms):** <http://libnoise.sourceforge.net/examples/worms/index.html>

**More feedback for stuck/incapacitated entities:**

* Write a shader to grayscale it? Or add a diagonal striped pattern? And/or animate that?

**Pause/Game over menu:**

* I wanted to add actual spiders crawling over your screen (one top, one bottom/diagonal). Those are the ones that create new lines and shoot the menus out of that.

**Smarter/more work around home bases and resetting players:**

* Prevent computer entities from visiting your home base?? (They’ll never enter an edge if the opposite side is a home base? And flying ones steer away from home bases?)
* Cap computer at 9, players at 19?
* If no larva are on-screen, always spawn a new one? (If players exist with few points?)
* Or, maybe, the terrain for edges around a home base are *fixed (and positive)*? They are always “free jump”, “shield”, etc.

**Further control web generation** ( = smooth out very rare bugs)

* **Removing entities from edges** (when they die) => this fails, on rare occasions, but I don’t know why.
* **Sometimes, points/edges do still overlap existing edges.** (If their distance *just* fell short of the edge, but also wasn’t enough to snap them to it.) Fix this? (Do one pass where we collide all points with edges, then move them away if so.)
  + Anytime an edge is changed, detect all points within its radius.
  + For all of those points, detect if they overlap an edge (they’re not connected with). If so, either push them out, or connect them.

**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.