NeoLemmixSharp Design Doc

Introduction

NeoLemmixSharp is an in-development engine for lemmings. The engine is written in the C# language (C# 13, DotNet 9), and uses the MonoGame framework.

The name “NeoLemmixSharp” is a temporary name, to be changed in the future (suggestions are welcome).

Aims

NeoLemmixSharp is designed to be a “superset” of the features of regular NeoLemmix. The engine is intended to be backward compatible with NeoLemmix, aiming for 99.9% compatibility. That is, the intention is that the vast majority of existing NeoLemmix levels should play identically in the new engine, and the few that do not are hopefully easy to tweak.

In addition to the features expected of NeoLemmix, a greatly expanded feature set is planned. These features draw inspiration from various sources, including lots of concepts from Lemmings Revolution.

Design Philosophy:

Lemmings was initially released for the Amiga in 1991 by DMA Design. Lemmings was designed to be a puzzle-strategy type game. Three decades later, the game’s legacy endures amongst members of the Lemmings community. Within this community, levels are generally created with a puzzle-oriented design. The numerous quality-of-life changes present within engines such as NeoLemmix complement this philosophy. As such, one aspect of the original games that no longer has much presence is that of the execution difficulty – where the player must apply quick reaction times to guide the lemmings toward their goal.

Although this engine also heavily leans towards the precision puzzle aspect, the execution challenge aspect still has a place within the game. Therefore, like the SuperLemmix engine (developed by WillLem), support for execution-based challenge is planned.

The NeoLemmixSharp engine is being built with some key principles in mind:

- All information must be easily available to the player upon first glance. A general community guideline is that objects hidden behind terrain is frowned upon. This is more of a level creation guideline, but the principle still applies to the engine design.
- A lemming's characteristics must be easily identifiable - permanent skills, neutral, zombie, team affiliation (see below).
- The effects of gameplay objects must be obvious, or easily discoverable (the player hovers mouse over and extra information is displayed).

- All interactions must be lemming based: The player must never be able to interact with the level unless it is by assigning a skill to a lemming. The sole exception to this is when the player may modify the release rate of a hatch.

Therefore, careful consideration must be given to the visual design of the planned features of NeoLemmixSharp. See the section on Lemming Sprites for more details.

Gameplay changes

While most mechanics and functionality from NeoLemmix are intended to be preserved in NeoLemmixSharp, there are necessarily going to be changes to certain aspects of gameplay.

One change involves the hitbox of lemmings, regarding gadgets such as traps, water, exits, and so on. In NeoLemmix, a lemming's hitbox consists of a single pixel, which is usually located beneath the lemming's feet. This pixel, hereby referred to as the "anchor pixel", is used for both terrain and gadget collisions. In NeoLemmixSharp this anchor pixel will remain and will be used in the same manner. However, in addition to this pixel, there will be an additional "foot pixel" that counts towards gadget checks only. This change is necessary for implementing features such as multiple gravities, and the expanded gadget system.

The foot pixel will generally be positioned one pixel above the anchor pixel (relative to the lemming's personal gravity). Exceptions to this are the climber and slider states, where the foot pixel "sticks out of the wall" by one pixel. In all cases, the anchor pixel and foot pixel are adjacent.

In terms of maintaining backwards compatibility with NeoLemmix, when a level from that engine is loaded in NeoLemmixSharp, all hitboxes will be automatically reduced in height by one pixel. This change will be functionally identical to previous behaviour.

A consequence of this engine difference is that when a user creates a new style, the hitboxes are no longer required to extend one pixel beneath the gadget sprite (in the case of water tiles, for example).

A note on lemming sprites

The basic lemming sprite is small by design, with three distinct colours for hair, skin, and body, and utilising few pixels to render. The NeoLemmixSharp will expand on this design with two new features: Lemming teams, and acid and water lemmings.

One planned feature is that of different coloured lemmings (teams). This includes different coloured skills in the skill panel. The skill panel of a given level may feature multiple instances of the builder skill, coloured according to the corresponding team it may be assigned to. There is also the question of "team neutral" skills, which may be assigned to any team without restrictions.

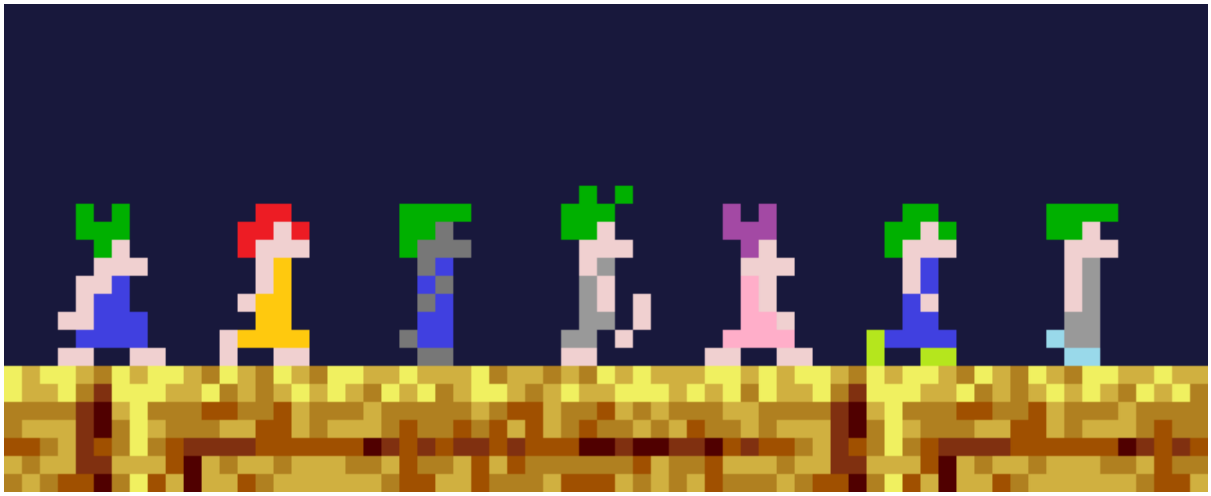
To ensure visual clarity, special consideration to lemming team colours must be considered. This includes accommodating for things such as colour-blindness in players. Team colours must be chosen carefully to maximize visual differences. To reduce the potential for visual confusion, the number of teams is restricted to a maximum of six per level.

Consider all the permutations of lemming state that must be visually communicated to the player:

1. Swap hair and body colours – this lemming has a permanent skill.
2. Grey body – this lemming is a neutral
3. Grey skin – this lemming is a zombie.

These are all the possible permutations for a lemming in NeoLemmix. In this proposed engine, there would also be:

4. Team colouration – instead of the classic green/blue, assorted colour pairs would represent different teams. For example, a lemming with a purple body and pink hair.
5. Acid/water lemmings – The feet of the lemming are coloured green/blue respectively, independent of hair/body colouration.



Above is a mockup of the various colour permutations possible within the NeoLemmixSharp engine. From left to right we have a classic lemming, a lemming from a different team (red/yellow), a zombie lemming from the first team, a neutral lemming from the first team, a lemming from a third team (pink/purple), an acid lemming from the first team, and a neutral water lemming from the first team.

When creating a level, the user may also choose multiple different sprites to appear in the same level. For example, Team A may use the classic lemming sprites, whereas Team B might use the Lemminas sprites, created by WillLem. Since these sprites are visually distinct, this will aid in readability.

Other Planned Features

- All features of NeoLemmix as standard. Backwards compatible – most (>99.9%) NeoLemmix levels should be openable in this version and should play identically.
- Config Options: Ability to disable gameplay helpers such as Pause, Replay/Rewind, Fast Forward (Classic mode). These options may also be enabled/disabled on a per-level/per-talisman basis.
- A colourblind mode will also be added, especially considering the differently coloured lemming teams (See above). This will essentially be an override system for lemming

colours. Players can choose from presets or can choose their own palettes to maximise visual clarity.

- Timed bombers/stoners. Both timed and instant bombers are supported, and the player can select a default behaviour for assigning these skills. Pressing the ALT key and assigning a bomber will choose the other option. This will allow for players that prefer timed bombers to utilise the precision of instant bombers as and when they choose. In addition, talismans may enforce timed bombers for extra challenge.
- Different gravities: Lemmings can walk up walls and fall sideways or upwards. New level objects would exist to change the gravity of a lemming that enters its hitbox. There would also be new skills to assign a specific gravity to a single lemming. Given the proposed support for multiple gravities, this opens up the possibility for interesting new interactions with different skills and gadget types.
    - One-way arrows: Perhaps the most logical extension. Arrows that oppose the direction of a destructive skill will negate that skill. Bombers will be unaffected as usual.
    - Blockers (and the one-way force direction fields): A blocker can only affect a lemming with the same or opposite gravity. Perpendicularly oriented lemmings are completely unaffected by the blocker field. Example: a blocker has its gravity pointing down (as normal). A lemming with gravity pointing up (walking on the ceiling) will be affected by the blocker in the obvious manner. A lemming with gravity pointing left (walking on the wall) will completely ignore the existence of the blocker. This includes being able to fall (moving horizontally) through the blocker's hitbox without being affected.
    - A feature that is technically supported is terrain that is only solid for lemmings with a certain gravity. That is, if a lemming falls vertically downwards, it may stand on a piece of terrain. But if a lemming falls vertically upwards, it passes through the terrain as if it was not there at all! This has the potential to be extremely difficult to convey to the player, so support for this may be dropped...
- Level wrapping: Both horizontal and vertical wrapping may be enabled independently. The level will also visually repeat.
- Different coloured lemmings (teams) (See above for more details).
- Acid and Water lemmings (See below for more details).
- Greatly expanded level objective system. In addition to the standard “Save X Lemmings” objective others will be available, including:
    - Kill All Zombies
    - Kill All Weasels (See below)
    - Kill All Lemmings
    - Level Trigger (e.g. certain level gadgets which can trigger a win if conditions are met. This can even be something zany like "feed 10 lemmings to a chameleon"!)
    - Level objectives may be combined, such as "Save 50 lemmings and kill all zombies". The level will not end until all criteria are met.
- Enforceable talismans: The level is played with the specific talisman restriction in effect (e.g. a talisman that is "Complete without using climbers" has the level start with no climbers available). The level will not be winnable unless the talisman is satisfied - this includes the level not ending if any zombies remain with the Kill All Zombies talisman enforced, etc.

- Weasels: The classic villains from Lemmings Revolution return! These enemies behave like lemmings, wandering back and forth. If they meet up with any lemmings, the weasel will kill them (Unique boxing glove animation? Lemming gets punched up into the air?). Weasels are always floaters and can never have any other skills assigned. Weasels are completely immune to the dreaded zombie virus and will even kill any zombie lemmings they encounter.
- Hatch Groups: Multiple sets of hatches may be linked together with independent release rates.
    - Example: a level has three hatches. Two are grouped together, with a lemming being dispensed from each in sequence. These hatches share their own release rate - setting it to 99 would alternate between the two at a fast rate. The third hatch of the level would be in its own group (a group of one). This hatch would have its own independent lemming count and release rate.
    - Levels which have this feature (that is, more than one hatch group. Levels with a single hatch group would be fine) would necessitate a change to the UI: The +- release buttons would disappear from the control panel. Instead, when the player hovers over a hatch, the +- buttons appear above it, as well as displaying the release rate (this visual indicator would also apply to all other hatches in the same group - hovering over one causes the indicator to appear over all other hatches etc). This is similar to how Lemmings Revolution operates.
    - Custom max and min release rates: A hatch group would be allowed to set the max and min release rates, ranging from 1 to 99. A level may start with a hatch at RR 50, which can be decreased to 20 and increased to 80, for example. Setting the max and min release rates to the same number has the same logical effect as locking the release rate.
    - Backwards compatibility: Levels from NeoLemmix are counted as having a single hatch group containing all hatches. This would therefore behave identically to previous behaviour.
- Fast forward lemmings: Once again, like the object found in Lemmings Revolution which speeds up lemmings for a short duration. This would be available as both a level gadget and a permanent skill. A lemming under this effect acts as a super lemming, moving 3x as fast as regular lemmings.
- Score system: In various console ports of lemmings (SNES), there was a simple score system in place. A similar system would exist for this engine. Ideas for this concept include:
    - For each lemming saved, 1000 points are awarded. For each skill not used, 100 points are awarded. For each second elapsed, subtract 1 point.
    - Score values are to be determined for the best balance.
    - In addition, certain level gadgets may also be able to trigger score differences - e.g. a lemming collects a gold ring (as in the various Sonic the Hedgehog styles), that awards 100 points.
    - New talismans could exist for score values upon completing a level.
    - There may also be penalties: each use of the pause button may incur a -10 point penalty. Perhaps rewinding will cost 50 points? This will have to be experimented with.
- Vastly expanded gadget system. This is very complex and deserves its own section!

New Gadget System

- Data driven – style creators can specify behaviours for gadgets in a diverse manner.
- The ability to add behaviours to gadgets within levels. Examples:
    - A trampoline sprite with the behaviour that any lemmings that fall into its hitbox become jumpers.
    - A button that requires a disarmer to activate, which turns off a flamethrower trap for 10 seconds. It also removes the disarmer skill from that lemming!
    - It would even be possible to have a trap that, upon killing a lemming, adds +1 minute to the level timer, removes 2 builders from the skill panel, plays a custom sound, and sends a signal to another gadget to cause some water to rise.
- Moveable/expandable gadgets: level gadgets (hatches, exits, traps, water, etc) may be assigned movement options. Gadgets which support resizing/stretching (such as water) may grow and shrink in the directions that the sprites allow. For instance, water that rises and falls, similar to the water gadgets in Lemmings Revolution. This also includes stretching horizontally as well as vertically. Gadgets may also move around in the level. There will be no random elements - all movement will be specified by the level creator.
- Switches and triggers: Similar to those found in Lemmings Revolution. These gadgets would be linked to other gadgets, such as enabling/disabling traps, causing water gadgets to fill up or drain a space enabling/disabling hatches, exits, etc. Weasels would be able to activate such switches, but zombies would not. Switches would include the "Green Lever" gadget which is directional and reversible, and buttons. Buttons would be reused from existing NeoLemmix styles and would act as one-time triggers. This would also give more utility to the disarmer skill: Certain mechanisms may be triggerable only by a disarmer. (The disarmer skill may have to be renamed to something like the "tinkerer" skill if this is implemented.)
- New Gadget: Metal grate (better name pending?). Similar to the gadget found in Lemmings Revolution, this is a piece of metal (indestructible terrain) which can be toggled on or off via a switch. When off, lemmings can pass through it as if it was not there. If a lemming occupies the same space as the hatch when it is activated, that lemming is killed similarly to Lemmings Revolution. QUESTION: In Lemmings Revolution, builders that collided with it would turn around similarly to a blocker. Would this behaviour exist here? Or would the grate simply cancel the builder as if it hit a wall. There are also questions raised about adding terrain while the grate is off: should this be possible? If so what would happen to the terrain when the grate is switched on?
- New Gadget type: Grinder (better name pending). Lemmings Revolution features a spinning blade gadget that can move along a predetermined path. This gadget had the ability to destroy any terrain it encountered and was deadly to lemmings and weasels alike.
- Custom lemming sprites for gadget interactions: In addition to the standard animations, custom lemming sprites may be added which a style's gadgets may point to when animating a lemming. This includes (but is not limited to) lemming exit sprites, lemming death sprites, teleport sprites, and so on. Drowning may have to be treated separately, as that has special interactions with swimmers, etc. Native support for Lemmings 2 exit animations is also planned (lemmings crawl into Polar's igloo, dive into Sports' golf tee, drunkenly stumble into Highland's telephone box, etc).
- Logic gates: Gadgets may have inputs and outputs. An output may be linked to a number of inputs and send them a signal when certain conditions are met. Logic gates such as

AND, OR, NOT and XOR will be supported for the joining of logic signals. For example, two buttons connected to an AND gate, which is connected to a metal grate. Both buttons need to be pressed to open the grate.

- Updrafts: These gadgets will have updated interactions with lemmings that are falling in some manner - the faller, floater and glider.
    - Faller: If a lemming falls into an updraft which is oriented against the direction of gravity, that lemming's fall is slowed by the updraft and its fall distance is reset, potentially saving it from splatting. This is the standard behaviour as in NeoLemmix.
    - On the other hand, suppose a lemming falls into an updraft which is pointing in the same direction as its gravity, then the survivable fall distance is halved! The lemming will fall faster to indicate this problem.
    - Floater: If a floater enters an updraft that is pointing left or right relative to its gravity, that floater will be blown left or right accordingly. The floater will move in the same manner as a glider does ordinarily. The floater's facing direction will not change, only its motion.
    - If a floater enters an updraft which is pointing in the same direction as its gravity (pushing it down instead of pushing it up), then the umbrella collapses! The lemming becomes a faller, and is susceptible to splatting - The floater loses its umbrella for the duration of the fall, and can only start floating again after it has landed. The floater does not lose its permanent skill, just a temporary disabling.
    - Glider: If a glider enters an updraft that is pointing left or right relative to its gravity, the glider will either be hindered or helped by the updraft. If the updraft points opposite to the direction of motion, the glider will be halted in its horizontal motion - it will glide vertically down like a floater does. If the updraft points in the same horizontal direction, the glider will fall at a slower rate: 2 horizontal to 1 vertical, as opposed to the 1 to 1 motion as normal.
    - Like the floater, an updraft that points down will cause the glider to fall without glider.
    - Combine the glider, updraft, and the new ability to have moving gadgets, then this could lead to some very interesting interactions with moving updrafts!
- Holdable gadgets: A new kind of gadget is one that can be carried. Imagine a key located in a level, and a lock located elsewhere.
    - The key will be affected by gravity and will fall if the terrain beneath it is removed. These gadgets will also have their own gravity and fall in the respective direction. If the gadget falls out of bounds, it will reset to its original spawn position. However, obstacles such as fire or liquids will not reset the gadget.
    - A lemming can walk up to the key and begin carrying it. In this state, a lemming cannot have any skills applied to it, the lemming moves at half the normal speed (1 pixel every 2 frames), and the lemming’s agility is greatly reduced. That is, any small ledge that would cause an ordinary lemming to enter the “ascender” state, instead acts as a wall to a carrying lemming, causing it to turn around.
    - If a carrying lemming enters the “faller” state, its carried item will be dropped and will fall to the ground!
    - If a key falls and lands on top of a lemming, that lemming will automatically stop what they are doing and begin carrying the item, even if the lemming is a blocker!

- These gadgets can have different properties, such as “fragility”. That is, there is a safe fall distance for these gadgets, and beyond that distance the gadgets will “break” and be reset.

Regarding Water Lemmings and Acid Lemmings:

Lemmings Revolution introduced lemmings with the special ability of being able to walk upon liquids as if they were land. This feature is planned for the NeoLemmixSharp engine, with some considerations as to functionality.

Firstly, the acid and water lemmings will be available as an assignable skill. These two states, along with the preexisting swimmer skill, are all mutually exclusive. That is, a lemming may be assigned at most one of these three skills within a level. This is similar to the mutual exclusion between the floater and glider skills. In addition to the assignable skill, lemmings may be preplaced within a level with these skills. Hatches may also dispense lemmings with these skills preassigned as expected.

There will be three separate categories for liquids: water, acid, and lava. In order to preserve backwards compatibility with existing NeoLemmix levels, all three of these liquid types will behave identically for the swimmer skill. There are no special behaviours between the different liquid types, except for the interactions with acid/water lemmings.

Water lemmings will treat their respective liquid type as completely solid and indestructible. This includes having skills interrupted by water as if it was steel (builders will stop building, miners will stop mining, etc). In addition, climbers and sliders will treat the sides of water as solid, allowing for vertical movement identically to solid terrain. For a water lemming, the other types of liquid are hazardous, causing the lemming to drown. Finally, water lemmings may fall from any height onto a water tile and be immune to splatting. Of course, the same thing applies to acid lemmings with their respective liquid type.

Thus, the lava type will always be hazardous to an acid/water lemming, especially since those lemmings cannot be assigned swimmers.

In order to accommodate this separation of liquid types, existing styles will need to be updated to take advantage of this new system. Amongst the original styles, obvious choices can be made: the green goo from orig_marble can clearly be marked as acid, and the lava from orig_fire is exactly that. The rest of the styles use water directly.

However, the distinction is less clear for other liquids, particularly those from the ohno_brick, ohno_rock, and ohno_bubble styles. What type of liquid these should be categorized as is unclear. As mentioned above, for the purposes of backwards compatibility with NeoLemmix, this makes little difference. For the purposes of the NeoLemmixSharp engine, decisions will need to be made. This can be put up to community vote, if enough people are interested.