



An educational impact game about vector-borne diseases

Team

Design	Sabine Neuser	sabine_franziska.neuser@smail.th-koeln.de
Design	Kilian Schmitt	contact@kilianschmitt.de
Code	Arthur Sliwka	arthur.sliwka@smail.th-koeln.de
Arts	Lisa Besse	lisa_jane_marie.besse@smail.th-koeln.de

Overview

Project Scope

- Platform: Browser, WebGL
- Genre: Impact Game, Educational Game
- Target: 13-16 year old European students in High School
- Impact: Awareness about vector-borne diseases and their spread
- Time: 6 Weeks for MVP

Elevator Pitch

InVector is a browser-based **educational impact game** targeted at European high school **students in 7th to 9th grade**, which will be used to enhance a school curriculum. The game aims to raise **awareness about major vector-borne diseases** that are at risk to spread into Europe, thus educating about the diseases and enabling the users to **adapt their behaviour** accordingly.

Project Description

InVector consists of several educational minigames centered around the topic of vector-borne diseases, as well as a **disease constructor**.

Completing/playing the minigames or completing specific challenges will unlock disease-specific **traits**.

With these traits, an artificial vector-borne disease may be created in the disease constructor -- either to target a specific reason or in a "free build" mode.

Once the constructed disease is finalized, it will be compared against real, existing vector-borne diseases, resulting in an **information dump** containing statistics and text about the highest-scoring disease of the comparison; and information on its spread, dangers and preventative measures.

It may also contain a short, fictional scenario of what the created disease would do in the real world, if it would be let loose as to support the **ludic setting**.

Setting-wise, the user will be treated as the apprentice of an evil scientist in their disease lab educating them about the diseases, encouraging them to create a strong disease while learning about actual disease prevention methods in the process.

To keep students interested in the game during the lesson, a big emphasis needs to be placed on **usability** and **framing** of the game content.

Art is intended to be kept friendly, accessible and minimalistic fitting for the young, mixed-gender audience, combining ludic and educational elements.

The estimated timeframe for a playthrough is **1-2 school lessons**, as it is intended to support a curriculum teaching diseases and health.

If possible within the timeframe of production, the involvement of teaching staff in the play session (by setting tasks or challenges for the class) is a goal that would help to individualize the lessons.

Influences

- Plague Inc. by Ndemic Creations
 - Playing with the goal to create a deadly disease
 - Genes and Traits to enhance disease
- Crusader Kings II Ruler Designer DLC by Paradox Development Studios
 - Heavily UI based game
 - Multitude of different traits and stats
- Battle Kid by Amnesty International
 - Character Creation as main game
- Bartender: The Right Mix by LandSecurities
 - Combine elements for different outcomes
 - Comparison and rating against predefined, existing drinks

Design Specifics

Narrative

The narrative design of InVector is kept on the lower side of things -- there are no quest or elaborate stories to be told. Narrative is only used to set an overall vibe of the game.

The game will be framed and themed around an **evil scientist**, who acts as the mentor of the user (apprentice).

Both the tutorial and the ludic approach to the serious topic of diseases can be conveyed in an engaging way to the young, mixed-gender audience of the game.

The scientist will greet the player, and guide them through the first steps and overall rules of the game.

After submitting their disease, he will reappear and “judge” the creation, and he will be the one **presenting the information** and possible scenarios to the user.

A unique feature used in the game to convey the narrative is the **communicator**.

This device is placed under the “game” screen, and features a sprite showing the scientist in different poses, as well as space to display text.

The communicator is used to make the scientist graspable and visualize him as a character in a game that, in itself, doesn't offer space for interactable NPCs.

Whenever a tutorial, game rules or hints need to be shown to the player, the communicator serves as a bridge between the Game Designer and the Player.

The goal of using an “evil” setting with a mad scientist character is to create a dissonance between the gameplay and visuals and the serious topic.

It allows the players to approach the topic of potentially deadly diseases lightheartedly and freed from any moral dilemmas.

At the same time, it allows us to overcome potential reluctance to learn about “boring” topics like hygiene or disease prevention.

In a situation where the player can strengthen their disease against the effectiveness of hand soap, the player is first and foremost focused on improving the disease. In the background however, it reinforces the knowledge that washing one's hands usually protects from infections.

Minigames

In InVector, minigames are used as a means to unlock additional traits in the Constructor upon completing and gain additional knowledge about vector-borne diseases while playing.

Minigame #1 – “Bite the host”

For more details, please refer to the minigame GDD [here](#).

In this 2D minigame, the player plays as a **tick** crawling on the skin of a human host’s body and aiming to find the perfect spot to bite and feed. Operating within a **time limit**, and with the camera being at a macro close-up range to the skin, you have to find out where on the body the tick/avatar is situated and try to reach an area which has a low likeliness of the tick being found. Use then the “bite” action to confirm your choice.

Educational effect:

- **Learn about the hotspots** on the human body ticks favor to attach themselves to.
- Learn where to check on your body for ticks.

User skills:

- Visual orientation, recognition of micro-structures
- Adaptation to clumsiness and slowness in avatar movement
- Memory
- Strategy: Fast decision making (under time pressure)

Constructor

The constructor part of InVector is a feature in which a custom disease is built from a pool of available traits. These traits are **unlocked by completing minigames**.

A disease has 5 **Stats** (Infection spread rate, Diagnosability, Treatment Effectiveness, Progression rate of the disease , Severity) which will be used to rate its level of danger, as well as to relate it to existing diseases.

Traits affect these stats beneficially and/or maleficially.

Each trait has a **monetary cost**, which will be deducted from the player’s “budget”, which will be based against the pool of unlocked traits they have accumulated -- e.g. the scientist mentor will trust more funds to his apprentice as they learn more about infectious diseases.

This adds a shallow progression level to the game; and adds a limit to the traits that can be used, forcing the player to put more thought into the customization process of the disease.

Each trait also has a **hidden modifier for symptoms** on it. As soon as the added modifiers for a symptom cross the threshold of 1, the symptom will become active and visible in the disease. Symptoms are things like coughs, rashes, fevers, headaches and so on, and **can affect stats** like infection spread positively, but will also negatively raise the treatment effectiveness or diagnosability of the disease.

Symptoms cannot be purchased directly, and cannot be removed as long as they are held active by traits -- they're **organically added to the disease** as it evolves.

Costs:

- Balanced against resources -- money per construction process
- Amount is balanced against amount of unlocked traits
- When entering a construction process, the player has their budget and pool of unlocked traits they can use
- They can either construct something, or cancel the process and go back to minigame to unlock traits, and thus increase their budget
- Unlimited amount of traits that can be slapped onto one disease, just regulated by cost

Relation between Symptoms and Traits:

- Traits have hidden chances of several symptoms on them
Example: The "Ambiguous" trait has a 0.5 "Cough" and a "0.5" Fever symptom modifier
If another trait is added that also adds a 0.5 "Cough" symptom modifier, the symptom would become active/visible in the disease and apply its effects on the stats
→ as soon as a symptom modifier crosses the threshold of the value 1, it becomes active
- Symptoms cannot be bought directly, but also cannot be avoided (unless you choose your traits accordingly -- even then there may be traits that are so potent that there's already a symptom modifier of 1 on them)
- Thus there's no conflict for the player having to choose between symptoms and traits for costs, or for us having to add an artificial gate of "you have to choose at least x symptoms"
- Symptoms appear organically on the disease, making it feel more natural

Pathogens and Vectors:

- Separate slots for 1x vector and 1x pathogen, which need to be filled regardless of chosen traits

- Made so they're distinct from other traits, and not impacted directly by the costs

Golden Goals / Future Plans

- **Printing**

The feature to print out a fact sheet of the created disease and/or the existing disease could be both a rewarding factor for the students and a great way to deepen knowledge on the diseases and prevention methods.

These sheets can also be used as a basis of a discussion or question round for the conclusion of this lesson block on vector-borne diseases. It could also be possible for the teacher to give a homework like "Research and print out 3 real vector-borne diseases" to then talk about during the next school lesson.

- **Achievements or Challenges**

A feature that would incentivise replaying the game would be challenges along the lines of "Create a disease at least 75% similar to Malaria" or "Score at least 200 points in Minigame ABC".

Players could then also become "experts" in certain areas, teaching others in class or exchange specific findings.

- **Involvement of teaching staff in the playing process → directly in-game**

Opportunities to establish closed groups or rooms within a network would enable the option to designate one machine as a "teacher" computer.

Unique features for a teacher would include:

- Overviewing all results submitted by "student" computers
- Issuing special challenges to the students in accordance to the curriculum or current lesson

- **More minigames**

To enrich the player experience and to educate in more depth, a future version of InVector would feature at least 5-10 minigames, tailored to different vectors, pathogens and other factors inherent to vector-borne diseases. Each minigame would then feature a special learning goal like the above described and already implemented tick-minigame.

Art Specifics

Artstyle

The game will be displayed entirely in 2D and is almost entirely **UI-based**. The artstyle of InVector is kept minimalistic, with big and clear shapes. This has the benefit of being less taxing on the artist as asset production can be faster, as well as offering a **well readable, non-distracting** environment for the user.

Since a mobile port is also a goal further down the road in development, a simplistic artstyle will make it easier to adjust the game look for a mobile audience.

Colours

The game's colour palette is kept in light and friendly colours to keep it accessible to all users, and to make sure colours aren't distracting from the game content.

Colours are being used to indicate and group different traits in the constructor.



Interface

As InVector is mainly played by interacting with the game interface, great care needed to be taken to design the UI to be accessible and coherent.

While there are many game systems, it was important to not overwhelm the user with too much clutter.

To achieve and plan the layout, wireframes were created and iterated upon based on internal and external feedback.

Icons

The icons are made out of simple shapes and just one color in order to be more clean and readable.



Based on the principle of Form Follows Function, iterative surveys were sent out to testers to determine whether the icons were recognizable for the largest part.

With this method, testers have to describe the Form of the icon and note what they'd imagine the Functionality of the icon to be.

Any icon reaching more than 50% recognition is deemed final.

Technical Specifics

Platforms

The chosen primary platform is a web browser due to the advantages it bears for the usage in classrooms.

- Easily accessible via links or searches
- No installation necessary
- No issues regarding updates or outdated versions

Golden Goals / Future Plans

Another possible platform would be the mobile platform, which would enable a more customizable relation between the game and classroom, e.g. homework or even supporting “field trips” to museums or external workshops; since most students in InVector’s target audience will own a mobile device.

The nature of the game as a calm, UI based project would be perfectly suitable for a port.

Concerns here would be the non-standardized aspect ratios and different OS’s across all devices.

Evaluations

Gameplay

To evaluate the impact and engagement of InVector’s gameplay, live playtests were held to determine weak points that needed revision.

In addition to observing notes, testers were given a survey with a mix of prewritten statements to choose from, and space for own thoughts and criticism.

Online playtests were another option where survey feedback can be gathered, although with those there's a tradeoff between more, anonymous feedback vs. on hands observing notes and closer contact.

UX

As mentioned in the Art section, iterative testing concerning the usability and readability of the game are of great importance.

For the testing of icons, special surveys based on the Form Follow Function principle are used to gather independent feedback to determine and revise weak icons.

Regarding the overall interface, the above mentioned live playtests and demos with follow up surveys and feedback are best suited.

Golden Goals / Future Plans

To evaluate the impact on the actual target audience, playtests in schools with highschool students is absolutely necessary.

As the summer holidays unfortunately interfered with plans to contact schools and test with students within the project timeframe, this is something that would need to happen frequently in any further development of the project.