

Snake begins



Report by Ruben Nielsen - rube0731@stud.kea.dk
Game is live at: <http://goo.gl/iW4oH7>

[Introduction](#)

[The concept of the game](#)

[The current state of the game](#)

[The development process](#)

[Scalability](#)

[Design](#)

[Graphics and Sounds](#)

[Player motivation](#)

[Monetization](#)

[Ustertests with focus groups](#)

[Literature](#)

Introduction

An introduction to the project, and why we chose it

First of all, if you haven't tried out the game yet, go do so at <http://goo.gl/iW4oH7>. The game is a modern adaptation of the old, well known Snake game; popular on cellphones from the era before smart phones. While brainstorming for game ideas, we - in the group - agreed that there is a sore lack of proper snake games for smartphones. On the iOS store, there are a couple of direct ports of the old game, and a few who try to modernize the game, but the ones I tried were lacking in entertainment value or simply bugged to the point that they cannot be played.

What we want to accomplish with our game is to gather up everything that is snake and put it into a single game, available on PC and mobile. Traditionally in Snake, the player controls a snake by moving in four directions (up, down, left, and right) while trying to eat mice to grow longer; the goal of the game being to grow as long as possible. That is it. We want to add a lot of extra possibilities. More of this in the following section.

The concept of the game

A detailed explanation of what we want the game to be

As stated above, we want the game to include all that is Snake. More specifically, we want players to be able to choose which rules to play by. First of all, we wanted to give the player the option to choose how the snake should move:

- Traditional movement: up, down, left, and right
- Omnidirectional: The snake turns by degrees

These options add a lot to the experience. People who played the original Snake game may want to use the traditional movement to have an experience similar to what they remember from back then. Other players may find the omnidirectional movement more appealing. The next option we want to provide is singleplayer and multiplayer modes. In singleplayer mode, the player plays either for leisure or to break a previously set highscore. Multiplayer mode will have two sets of rules: either the two players compete to be the "last snake standing", trying to make the other player die; or, the players work together to get a highscore together. The last ruleset will be a test of cooperation and possibly a test of friendship for the players, since the game would end as soon as one of the players die, which should add to a tense atmosphere.

Lastly, in the competitive multiplayer mode, an option could be chosen to turn the game into Tron¹ instead of snake, so that the snakes keep growing and the mice are no longer in the game. This would add a very different experience to the game.

¹ <http://www.classicgamesarcade.com/game/21670/tron-game.html>

The reason for making all these options available to the player is twofold

1. We want to add as much diversity to the arcade-ish experience that is snake.
2. Many people have already tried snake games, and we want to cater to all of them, regardless of how they are used to playing the game.

The image below is a mockup of how we expect to let the player choose the options. Once the player has selected single-/multiplayer from the main menu, the following screen will show the options that make sense for the game type, and the player will be able to select what he wants and then start the game.

Movement

☒ Traditional

☐ Omnidirectional

Game mode

☒ Snake

☐ Tron

Movement speed

—|—

Level

Start

In addition to these various rules, we want to provide multiple levels to choose from. The default level is just a plain map with borders and nothing else. Other levels could have some geometry, which the snake will have to avoid. Levels would give the player a more varied play experience.

Aside from the game features, we want the game to be available on mobile platforms. The game is ideal for this platform, since the player only has two controls - left and right - so the screen won't be cluttered with buttons.

The current state of the game

A quick sum up of what is done and working

In the game's current state, we have not been able to provide all the features described above. We have not had the time and resources to port the game to mobile, and the PC version does not provide all the play types described. As such, the game should be considered a prototype, and a proof of concept. It delivers the basic experience in singleplayer and multiplayer, and it displays the dynamics of the game as well as the desired aesthetics. There is only one map in the game at this point, though the concept of levels is there in the sense that there is a single- and multiplayer setting, *and* the difficulty of each game is increased every time the snakes grow.

The development process

An explanation of our process and the tools we used

We decided to start the programming work from the main menu screen and forward from there. Since the `createjs` framework calls the root container for elements the “stage”, we decided to create “scenes” for each screen we needed. You can find them in **lib/scenes**. We started working on the `MainScene` for the main menu, and quickly decided that we needed a generic way of displaying GUI² elements like buttons. To that end, we created the **lib/ui** module. We use proper OOPD³ to structure our code, which means we separate our code into modules that make sense, and we create classes to contain our code (in javascript, classes are functions that can be instantiated with the keyword “new”). We created the `Button` class to easily setup our GUI.

Once the main menu was done, we started working on the singleplayer game in the `GameBoardScene`. There were some challenges associated with getting the snake to move as intended. The final solution was to keep the heads move history in a list, and have each body part use the history to know how to move. The math and logic is found in **lib/actors/snake.js:move()**. We have striven to comment on all logical blocks of code in the project, so even if the code does not make immediate sense, it should be possible to read through the comments and make some sense of it.

The only part of the project that does not use proper javascript classes is the `Button`. `Button` was supposed to inherit from **`createjs.Container`**, but when we couldn't immediately find documentation on how to call the superclass constructor, we decided to just make it a method that returns a **`createjs.Container`** object.

Once singleplayer was done, implementing multiplayer was trivial. We added a second player, and made the checks in **lib/scenes/gameBoard.js:onTick()** include `player2`.

² Graphical User Interface

³ Object Oriented Programming and Design

For registering controls, we made a wrapper **lib/game/game.js:Key** to easily handle key presses.

The sounds in the game, we downloaded from FreeSound⁴, who provides a database of sounds under the Creative Commons License. The choice of sounds will be discussed in the Design section.

Finally, all of our graphical assets (except for the mice), we created ourselves using Inkscape⁵, an open source vector graphics drawing tool. Using vector graphics allowed us to export models with the desired size, so we did not have to scale bitmaps. This allows for clearer images without pixelation.

Scalability

Since we wanted the game to be portable to mobile devices, we made some legwork early in the process. We knew that we would need to accommodate a lot of different display types, so we could not just make the game work with one default resolution and aspect ratio. To that end we have made sure that most of our code behaves well with resizing. Not everything is as it should be, since we did not get to the porting process, however, the buttons in the main menu will not leave the screen if another resolution is chosen, and the game itself will behave well as long as the screen is in landscape mode, that is the resolution is wider than it is high.

Design

The design decisions that drove our project

It hasn't been mentioned explicitly yet, but our target audience is children between 10-15 - particularly boys. While the game should conjure some nostalgic feelings with young adults, we wanted the game to appeal to kids primarily. We were very much aware of the MDA framework - Mechanics, Dynamics, and Aesthetics. To appeal to kids, we needed the mechanics to be simple. It should be possible to sit down and start playing without any prior knowledge of the game. We have achieved this by letting the player press a single button to start playing, and then keep the controls very sparse. The dynamics too, should be simple to understand. Each game is relatively short in timespan, so we decided that the player should learn by playing. The first time the player hits a mouse, they will notice the snake getting longer and the score ticking up. The first time the player hits the borders of the screen or the snake's body, they will notice dying. Nothing complicated, and it is quickly learned. The aesthetic is built with the graphics and the sounds. We wanted to keep the tone light and humorous.

⁴ <https://www.freesound.org/>

⁵ <https://inkscape.org/en/>

Graphics and Sounds

The graphics are simple and cartoonish so as to not be distracting. The sounds were carefully chosen:

Background ambiance - Birds chirping in the woods. Idyllic and non-intrusive.

Sounds effects when eating mice - Chewing noises and burping. Sharp contrast to the idyllic ambiance.

We feel that the contrast between the sound effects and the ambiance provides a humorous tone for the oldest part of the target audience, and for the younger audience, the burping itself should provide humour on a lower level. We chose four different sound effects for eating mice, that are chosen at random each time, so the experience doesn't get too repetitive.

The importance of these silly sounds should not be underestimated. They are part of the reward system. We do have a score tallying for each mice eaten, but that is not very interesting for the younger audience. Instead they chase the funny sounds. As mentioned above, we have 4 sound effects. In detail, they are

- A not-so-loud chewing noise to indicate a mouse was eaten.
- A loud chewing noise for emphasis.
- A burp for comedy.
- A loud, deep, over-the-top burp which you can feel in your throat every time you hear it.

From this list, an escalation should be very apparent - even to the young audience. The young audience will chase the most excessive sound effect, because it is funny.

The last sound effect, which has not been mentioned, is the sound when the player dies. We chose a classic - the Wilhelm Scream⁶. The sound serves two purposes:

1. It clearly indicates that the player lost.
2. It adds a certain hyperbole to the otherwise annoying event of losing, which should add a smile to the player, hopefully making them want to try another round (if nothing else, then to hear some more burping).

Player motivation

We wanted the game to appeal to more than just one player type. Specifically, we considered the "Bartle play types⁷". The traditional snake experience appeals solely to the *achievers*, as the only reward type was a highscore. Our game has that and more.

We appeal to the *killers* by providing a competitive multiplayer mode and we appeal to the socializers by providing a cooperative multiplayer mode.

We did not discuss appealing to the *Explorers*, though we could do so by making levels and possibly some powerups unlockable, that is, lock off some secret features, which will be

⁶ <https://www.youtube.com/watch?v=Zf8aBFTVNEU>

⁷ http://en.wikipedia.org/wiki/Bartle_Test

available to the player when he reaches some predefined goals (win X amounts of multiplayer games; reach a highscore of X in level Y; etc.).

Monetization

In regard to the unlockables mentioned above, this also provides a platform for making the game a *freemium* on mobile devices. The popular term means that the game is free to play, but the player can pay to unlock extra content. So by this process, we could let players pay to unlock everything in the game, while player who do not pay will have to unlock content through playing. We have not discussed this kind of monetization in the group, and quite frankly, I despise the business model of *freemium*, but this would be a way to do it.

Usertests with focus groups

As the timespan of this project is considered rather short, we have not invested a lot of time in usertests, however, we did sit down a 10-year-old girl and a 13-year-old boy and have them play both single- and multiplayer. What this showed us was that the girl couldn't quite control the snake with the speed at which it was going. The boy had no issues controlling the snake, and he rather enjoyed the humorous tone. Luckily, this supports our considerations of the target audience and aesthetics, though two people would hardly be considered a proper test group.

Literature

- We used the hittest function from my asteroids game:
<http://casgamese20109.keaweb.dk/mini-project/> - Accessed Jan 16, 2015
- An example of the Tron game:
<http://www.classicgamesarcade.com/game/21670/tron-game.html> - Accessed Jan 15, 2015
- A website that provides sound effect free of charge: <https://www.freesound.org/> - Accessed Jan 13, 2015
- The website of the open source Inkscape project. A tool for editing vector graphics:
<https://inkscape.org/en/> - Accessed Jan 14, 2015
- A link to a compilation video of the Wilhelm Scream:
<https://www.youtube.com/watch?v=Zf8aBFTVNEU> - Accessed Jan 15, 2015
- Wikipedia's explanation of the Battle play types: http://en.wikipedia.org/wiki/Battle_Test - Accessed Jan 15, 2015