# Snake Remake

## Requirements

### User inputs

The user can start and pause the program by pressing specific keys.

Two methods one for start and one for pausing the game which react on keyboard inputs.

* Start: S
* Pause: P

The user can change the direction of the snake by using the arrow keys.

One method which handles the direction changes. It reacts on the keyboard inputs and makes sure that you are not able to go in the opposite direction.

* Left: ←
* Right: →
* Up: ↑
* Down: ↓

The user can change the color of the snake by pressing a key.

One method changes the color if it gets input from the keyboard.

* By pressing C the color changes from black to red.

### GameGrid

* One method defines initializes the game filed and defines its size (size is equal to borders for the snake movements)
* One method defines the color of the background and the snake.
* One method displays the movement of the snake on the gamegrid.
* One method creates Apples which pop up on random spots on the field (as soon as one apple has been eaten)
* One method displays the score on a top label and the running clock.

### Snake

* If the snake reaches the border or if it hits some point of its body the game is over

A method checks if the next keyboard-input is valid or not and stops the program if not.

* If the snake eats an apple it becomes longer.

A method adds body points if the head eats an apple.

* The snake moves at a certain speed without even without any input (with input it changes direction)

The method moves the body points of the snake further every “xyz” milliseconds in the current direction of snake.

* The speed of the snake begins to increase after a certain amount of time.

A timer method where at a certain time, the speed would increase.

### General

* One method counts the number of apples eaten and adds the points to a score.
* A clock method which keeps track of how long you have been playing.
* After game over, a method would print a statement of the user’s score and how long the user has survived in the game.

# Timetable and Milestones

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Task | due to | Content | Responsible person | status |
| **Requirements Paper** | 10/20 | Describe program requirements. | ?? (send email to Prof) |  |
|  | Describe functions and inputs |  |  |
|  | … |  |  |
|  | … |  |  |
|  |  |  |  |  |
| **Program-Design Paper** | ??? | Describe the design of the project implementation. The description should include English prose, UML diagrams and any viewgraphs necessary to elucidate the design. | ?? (put everything together in a nice IEEE-Format and send email to Prof) |  |
|  | Define structure |  |  |
|  | Define classes |  |  |
|  | Define methods |  |  |
|  | Define relationshiops between classes |  |  |
|  | Draw UML Diagram | Angi |  |
|  |  |  |  |  |
| Iteration 1: |  |  |  |  |
| **Subtasks for Iteration 1** |  | Read GameGrid Tutorials | Angi (selfstudy) |  |
|  | Try to run and install GameGrid class | Angi (selfstudy) |  |
|  | Try to create a game field | Angi (selfstudy) |  |
|  | Study the math method for random apple method | Angi (selfstudy) |  |
|  | Look over logic of program | Anthony |  |
|  | Study/code methods for snake body method | Anthony |  |
|  | Read Snake Tutorial | Anthony |  |
|  |  | Mike |  |
|  |  | MIke |  |
|  |  |  |  |
|  |  |  |  |  |
| **Iteration 1** | ??? | First round of implementing and testing. | ?? (send email to Prof) |  |
|  | Create a GameGrid field for our Snake program | Angi |  |
|  | Try to let an apple pop up on the field | Angi |  |
|  | Try to form the snake | Anthony |  |
|  | Make it be able to move around | Anthony |  |
|  |  | Mike |  |
|  |  | Mike |  |
|  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Iteration 2:** |  |  |  |  |
| **Subtasks for Iteration 2** |  |  | Angi |  |
|  |  | Angi |  |
|  |  | Anthony |  |
|  |  | Anthony |  |
|  |  | Mike |  |
|  |  | MIke |  |
|  |  |  |  |
|  |  |  |  |  |
| **Iteration 2** | ??? | Second round of implementing and testing. | ?? (send email to Prof) |  |
|  |  | Angi |  |
|  |  | Angi |  |
|  |  | Anthony |  |
|  |  | Anthony |  |
|  |  | MIke |  |
|  |  | MIke |  |
|  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Iteration 3:** |  |  |  |  |
| **Subtasks for Iteration 3** |  |  | Angi |  |
|  |  | Angi |  |
|  |  | Anthony |  |
|  |  | Anthony |  |
|  |  | Mike |  |
|  |  | MIke |  |
|  |  |  |  |
|  |  |  |  |  |
| **Iteration 3** | ??? | Third round of implementing and testing. | ?? (send email to Prof) |  |
|  |  | Angi |  |
|  |  | Angi |  |
|  |  | Anthony |  |
|  |  | Anthony |  |
|  |  | MIke |  |
|  |  | MIke |  |
|  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Final presentation** |  |  |  |  |
| **Subtasks for final Paper and Presentation** |  |  | Angi |  |
|  |  | Angi |  |
|  |  | Anthony |  |
|  |  | Anthony |  |
|  |  | Mike |  |
|  |  | MIke |  |
|  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
| **Presentation** | ??? | Presentation and Final Paper | ?? (send email to Prof) |  |
|  |  | Angi |  |
|  |  | Angi |  |
|  |  | Anthony |  |
|  |  | Anthony |  |
|  |  | MIke |  |
|  |  | MIke |  |
|  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |