# Dokument mit der stichpunktartigen Notation von Operationsaufrufen und Variablenwerten, welches aufzeigt was passiert sobald die Pfeil-Rechts-Taste gedrückt wird

* Durch die Betätigung der Pfeil-Rechts-Taste springt das Programm aus dem loop in den Event-Handler.
* Prüfung: welche Taste? ( key == OF\_KEY\_RIGHT) trifft zu
* Ausführung des Handlungsschrittes (audMan->moveBy(1,0)).
* audman ist ein Objekt der Klasse von AuDMan, und dereferenziert move by,

--> Zugriff auf Methode (void AuDMan::moveBy(int dx, int dy) )

* Bei Ausführung:

1. das Objekt level dereferenziert getWidth() -> Zugriff auf Methode (int Level::getWidth() )
2. Bei Ausführung dieser Methode wird der Wert der Variablen Width zurückgegeben
3. in der Variable int Breite wird nun der Wert der Variablen Width gespeichert

* dieser Prozess wird ebenfalls für den integer Höhe durchgeführt, in diesem Fall wird auf die Methode getHeight() in der Klasse Level zugegriffen und dessen zurückgegebener Wert von Height im integer Höhe gespeichert.
* in der Variable posX ein neuer Wert gespeichert, indem die Rechnung ((posX + dx + breite) % breite) vollzogen wird und dessen Wert nun in der Variablen posX gespeichert wird.
* Ebenso wird dies mit der posY durchgeführt, hier wird allerdings das Ergebnis der Rechnung ((posY + dy + hoehe) % hoehe) zum neuen posY Wert.
* 3 if- Anweisungen werden durchlaufen

1.

* das Objekt level der Klasse Level dereferenziert get field(poX,posY) , greift auf die Methode get field in der Klasse Level zurück und gibt im Verlauf der Ausführung posX und posY zurück
* Vergleich : stimmt einer der beiden Werte posX, oder posY mit 1 überein?
* Ja? -> (posX = posX - dx; und posY = posY – dy; ) wird ausgeführt Nein? Nächste if-Anweisung

2.

* das Objekt level der Klasse Level dereferenziert get field(poX,posY) , greift auf die Methode get field in der Klasse Level zurück und gibt im Verlauf der Ausführung posX und posY zurück
* Vergleich : stimmt einer der beiden Werte posX, oder posY mit 2 überein?
* Nein? -> nächste if-Anweisung
* Ja? -> ( das Objekt level dereferenziert set field, greift darauf zu und führt sie aus

--> if-Anweisung: Überprüfung der Bedingung: ( x>=0) && (x<width) && (y>=0) &&(y<height )

--> Ja? der Wert von value wird field (x,y) übergeben und der Wert des integers points wird mit eins addiert. Nein? Keine Ausführung der Anweisung)

3.

* das Objekt level der Klasse Level dereferenziert get field(poX,posY) , greift auf die Methode get field in der Klasse Level zurück und gibt im Verlauf der Ausführung posX und posY zurück
* Vergleich : stimmt einer der beiden Werte posX, oder posY mit 6 überein?
* Nein? Die Methode ist beendet und das Programm “springt” in die Main zurück
* Ja? das Objekt level dereferenziert set field, greift darauf zu und führt sie aus

--> if- Anweisung: Überprüfung der Bedingung: ( x>=0) && (x<width) && (y>=0) &&(y<height )

--> Nein? Die Anweisung wir nicht ausgeführt

--> Ja? der Wert von value wird field (x,y) übergeben und der Wert des integers points wird mit 10 addiert.

* Die move by Methode ist abgeschlossen, falls nicht bereits nach einer if-Anweisung geschehen, “springt” das Programm in die Main zurück
* Durchlaufen der if-Anweisung

if(audMan->getPosX() == ghost ->getPosX() && audMan -> getPosY() == ghost -> getPosY()) { audMan -> setPoints((audMan -> getPoints()-4));

* im void key pressed (da die Methode noch nicht beendet war)
* Überprüfung der if Bedingung(audMan->getPosX() == ghost ->getPosX() && audMan -> getPosY() == ghost -> getPosY())

--> hierfür wird getPosX von dem Objekt audMan dereferenziert,

--> auf getPosX wird zugegriffen und ausgeführt, der Wert der Variable posX wird zurückgegeben

--> getPosX wird von dem Objekt ghost dereferenziert,

--> auf getPosX wird zugegriffen und ausgeführt, der Wert der Variable posX wird zurückgegeben

--> die beiden Werte werden verglichen ( enthalten sie den gleichen Wert?)#

--> getPosY wird von dem Objekt audMan dereferenziert,

--> auf getPosY wird zugegriffen und ausgeführt, der Wert der Variable posY wird zurückgegeben

--> getPosY wird von dem Objekt ghost dereferenziert,

--> auf getPosY wird zugegriffen und ausgeführt, der Wert der Variable posY wird zurückgegeben

--> die beiden Werte werden verglichen ( enthalten sie den gleichen Wert?)

* Nein? Die Methode ist beendet und das Programm springt erneut ins Loop und führt fort
* Ist die Bedingung wahr(beide Bedingungen)

--> audMan dereferenziert setPoints, Zugriff und Ausführung der Methode: points wird der Wert von dem integer value übergeben

-->audMan dereferenziert getPoints, Zugriff und Ausführung der Methode: Rückgabe des Wertes von der Variabel points und Substraktion mit 4

* Nach Ende der Methode void key pressed (int key) springt das Programm erneut in den loop und führt dort fort, falls dies nicht schon bereits durch Nichterfüllen der if- Anweisung geschehen ist

Der angegebene Quellcode entspricht nicht dem Original, es wurde versucht auf diese Weise die Reihenfolge besser darzustellen, da Unsicherheit bei der Umsetzung bestand. Falls dies nur zur Verwirrung beiträgt, ggf. ignorieren. Weitere Erläuterungen hierzu im Testat.

void **keyPressed**(int key) {

If ( key == OF\_KEY\_RIGHT ) {

audMan->moveBy(1,0);

}

void AuDMan::**moveBy**(int dx, int dy) {

int breite = level->getWidth();

int Level::**getWidth**() {

return width;

}

int hoehe = level->getHeight();

int Level::**getHeight**() {

return height;

}

posX = (posX + dx + breite) % breite;

posY = (posY + dy + hoehe) % hoehe;

if (level->getField(posX,posY) == 1) {

int Level::**getField**(int x, int y) {

return field[x][y];

}

posX = posX - dx;

posY = posY - dy; }

if ( level->getField(posX,posY) == 2 )

int Level::**getField**(int x, int y) {

return field[x][y];

}

level->setField(posX,posY,0);

void Level::**setField**(int x, int y, int value) {

if ( (x>=0) && (x<width) && (y>=0) &&(y<height)) { field[x][y] = value;

}

points = points + 1;

}

if(level->getField(posX,posY) == 6 ){

int Level::**getField**(int x, int y) {

return field[x][y];

}

level->setField(posX,posY,0);

points = points + 10;

}

if(audMan->getPosX() == ghost ->getPosX() && audMan -> getPosX() == ghost -> getPosY()) {

audMan -> setPoints((audMan -> getPoints()-4));

}

int AuDMan::**getPosX**() const {

return posX;

}

int GHOST::**getPosX**() const {

return posX;

}   
  
  
int AuDMan::**getPosY**() const {

return posY;

}

int GHOST::**getPosY**() const {

return posY;

}

void AuDMan::**setPoints**(int value){

points = value;

}

int AuDMan::**getPoints** () {

return points;

}

Meilenstein Nr2; Gruppe 2