## a

## a

## a

### View

#### Felelősség

A megjelenítésért felelős osztály. Feladata hogy tartsa a kapcsolatot a felhasználó és a model között, grafikus megjelenést adjon az áramkörnek, melyről a vezetékek és kimenetek értékei leolvashatók.

#### Ősosztályok

Nincsenek

#### Interfészek

Nincsenek

#### Attribútumok

* **- JFrame frame:** grafikus elem, minden meghívott Draw() függvény ide rajzol.
* **-JButton playBtn, stopBtn, pauseBtn:** a futás állapotát szabályzó gombok.
* **-JButton stepBtn:** a léptetést kezelő gomb. Kattintásra azonnal egy darab *step* hívódik meg, a futás állapotától függetlenül.
* **-JButton loadBoard:** a filebetöltő párbeszédablakot megnyitó gomb
* **-JTextField timerIntervalField:** szövegmező, melyben megadható a léptetés gyakorisága, ms-ben.

#### Metódusok

* **+ buildVisual():** Az áramkör minden elemének digitalObject-jének kiszámolja az x,y koordinátáját, valamint minden wire-nek a töröttvonal reprezentációit. Mindezt a file beolvasás alatt megépített hierarchiaszintezés segítségével algoritmikusan végzi.
* **+ updateAll()**: kirajzolja az áramkör minden elemét. Az objektumokat hierarchia szerint kirajzolja, és az összes vezetéket is behúzza, értékének megfelelően.
* **+ updateWiresAndOutputs()**: Minden vezetéket és kimenetet újrarajzol. Azért van szükség erre a függvényre, mivel csak a vezetékek és kimenetek változnak az egyes pályabetöltések óta.
* **playBtnClick(event):** Elindítja a szimulációt. A pauseBtn és stopBtn-t engedélyezi, a playBtn-t tiltja.
* **stopBtnClick(event):** Megállítja a szimulációt, és az elemeket alaphelyzetbe állítja. A pauseBtn és stopBtn-t tiltja, a playBtn-t engedélyezi.
* **pauseBtnClick(event):** Megállítja a szimulációt. A pauseBtn és stopBtn-t tiltja, a playBtn-t engedélyezi.
* **stepBtnClick(event):** A futás állapotától függetlenül azonnal lépteti az áramkört.
* **loadBoardBtnClick(event):** Megnyitja az operációs rendszer fájlkezelőjét fájlkiválasztásra, majd ha az visszatér, megpróbálja betölteni a fájlban foglalt bhdl leírás alapján az áramkört.
* **timerIntervalFieldChange(event):** A léptetést ütemező időzítő intervallumát felülirja a timerIntervalField-ben található számmal, amennyiben az pozitív egész.

### Timer

#### Felelősség

Feladata hogy értesítse a Controllert a beállított paraméternek megfelelő időközönként, amennyiben a Controller elindította.

#### Ősosztályok

Nincsenek

#### Interfészek

Nincsenek

#### Attribútumok

* **- int tick**: Időzítéshez konstans mely a léptetési mértéket tárolja.

#### Metódusok

* **- void Tick()**: Számláló, jelez a **Controller**-nek amennyiben letelt a kívánt időköz (*tick*).
* + **void setInterval(int interval):** Beállítja a számláló időközét a paraméterként kapott értékre.
* **+ void start():** Elindíttja a számlálót
* **+ void stop():** Leállítja a számlálót.

### WireView

#### Felelősség

Lehetővé tegye és megvalósítsa az egyes vonalak kirajzolását

#### Ősosztályok

Nincsenek

#### Interfészek

Nincsenek

#### Attribútumok

* **- List<List> torottvonalak:** a töröttvonal a wire graikus reprezentációja, a tömbben a töröttvonal szakaszainak végpontjainakx,y koordinátáit tárolja. Mivel egy wire-nek több outputja lehet, több töröttvonalat kell tárolnunk.

#### Metódusok

* **+ Draw(int value):** A paraméter értékének megfelelően szürke vagy fekete színűen kirajzolja a töröttvonalakat a bemenetétől a kimenetéig

### Wire

#### Új Attribútumok

* **+WireView view:** fs

### DigitalObjectView

#### Felelősség

Feladata hogy az egyes DigitalObjectekhez nyilvántartsa a megjelenítéshez szükséges adatokat, valamint kérésre kirajzolja az objektumot.

#### Ősosztályok

Nincsenek

#### Interfészek

Nincsenek

#### Attribútumok

* **- int x,y:** Az elem koordinátái
* **- Image textura:** Az elemet beazonosító kép, mely a kirajzoláshoz kell.

#### Metódusok

* **+ void Draw():** edf

### DigitalObject

#### Új Attribútumok

* **+ DigitalObjectView view** : visszaadja az elem megjelenítéshez szükséges információkat tartalmazó objektumot.