

Documentation: Static View on the program



As the UML is large the links between the classes over the packages are not drawn. So the dependencies and are described in this part:

In **the Starter package** is the Main class which creates „TheApp“. „TheApp“ is responsible for the drawing actions and the user input. In addition it handles the point and level count.

For the drawing action it needs to call methods from the Controllable- and

InvaderComponentBuilder with help of the ComponentCreator. To handle the components events all Controllers are called. When every component moved the App calls the CollisionDetector.

To handle the user input we need to use the Direction-Enum.

The Builder Package is used to create all components which are drawn to the window. To start building them the one instance of the ComponentBuilder and the ComponentCreator are needed. The Builder is passed to the Creator which calls the builders method. The Builder creates the component which is responsible for building the corresponding view and model. The component, view and model are related one to one. The Bullet Component additionally has a reference to the model which created the bullet to be able to remove it from the model.

For handling the movement, updating the views, removing components and to make the player reach the next level or getting game over **the Controller package** is used. All classes / methods are static so every class may call their methods and to keep the components alive during the runtime. The specific controllers just handle the events. The abstract Controller is responsible for the other mentioned events.

Beside the controllers there is the CollisionDetector which is checking if any component overlaps or reaches the window's frame.

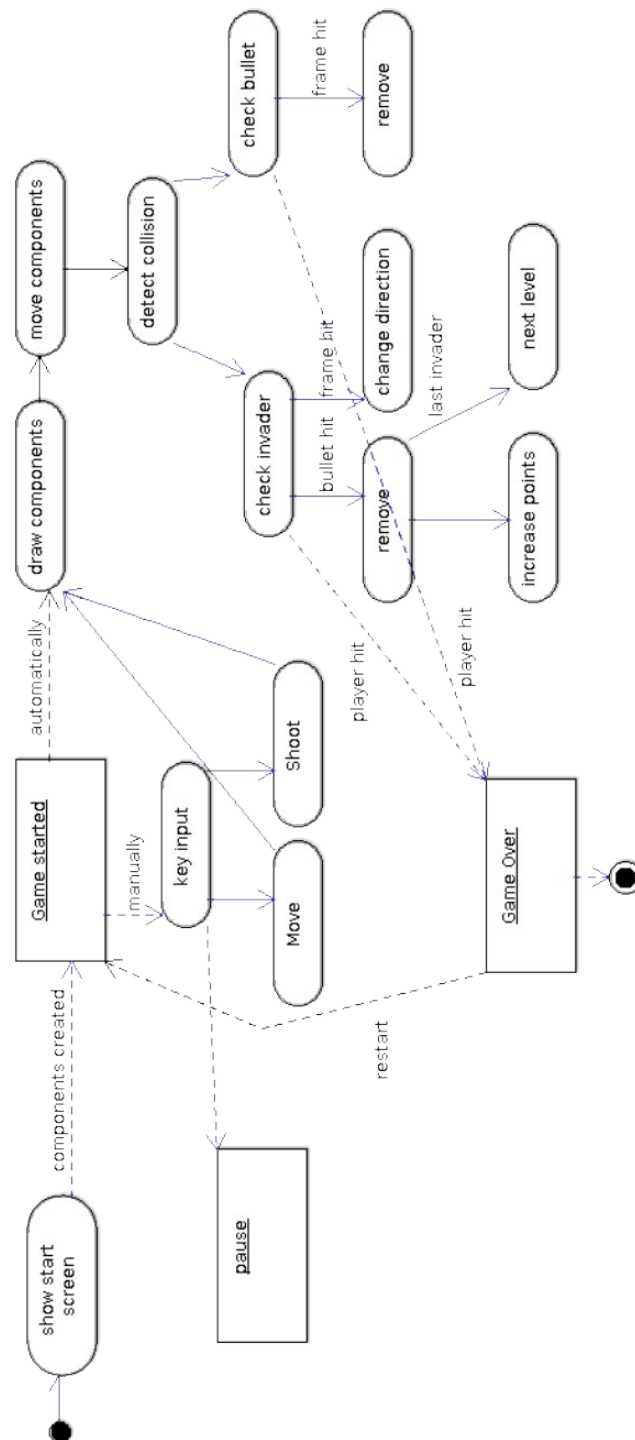
To draw the components we need **the Views package**. Here the Components get their shape and are drawn to the app. They have a one to one relation to the models.

The models in **the Model package** contain the size and the position of the component. They are the basis to the views and tell them where to draw them. The Invader and SpaceShip Model use the ComponentCreate with the BulletComponentBuilder to enable them to shoot. They are also referenced to the bulletcomponent. The cardinality is one model to n bullets.

Not models but only used for them are the Size and Position containing a value for the width and height / x axis and y axis.

The Enumeration package only contains the direction to tell the models in which direction they move. For Invaders and the player the directions are only LEFT and RIGHT and for bullets it is only UP and DOWN

Dynamic view on the components:



When the game starts the controls are displayed at the beginning. The User starts the game by hitting enter. Now the components are build by the componentbuilder (invadercomponentbuilder, controllablecomponentbuilder). It is fixed that there are six invaders created on the x axis at point zero and on the point 50 on the y axis. The player is create on x 350 y 650. Both components have the size 5/5 and are displayed as triangle.

„The App“ has a draw method which is called in a endless loop displaying and increasing the time and displays the level and points. Of Course this method makes the invaders move and check the collision.

The Invaders move automatically this way depending on the collision. When they hit the side frame they change the direction from left to right or the other way round and they move 50 points on the y axis toward the spaceship (from top to bottom). During the movement they randomly shoot a bullet (random number from 0 to 1 when it is less than 0.005 or higher than 0.995). If an Invader is hit by a bullet the number of points increases. If it was the last Invader the next level starts. Here the Invaders are duplicated in an addition row. During the collision detection the bullets are checked if they hit the frame and get destroyed if so. It is also checked if a bullet hits the player and makes the player game over.

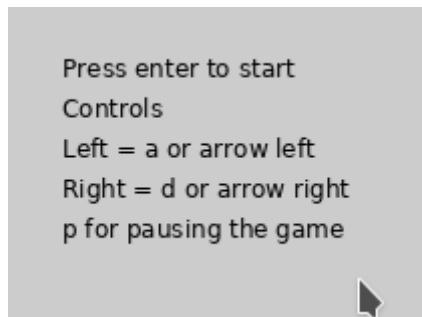
The spaceship moves when the user presses a or d or left or right arrow key. The player can shoot when he presses space. Pausing the game can be done by pressing „p“.

Invaders are restricted to one bullet until the bullet is destroyed (hit frame) and the player is restricted to two bullets until they get destroyed (hit an invader or the frame).

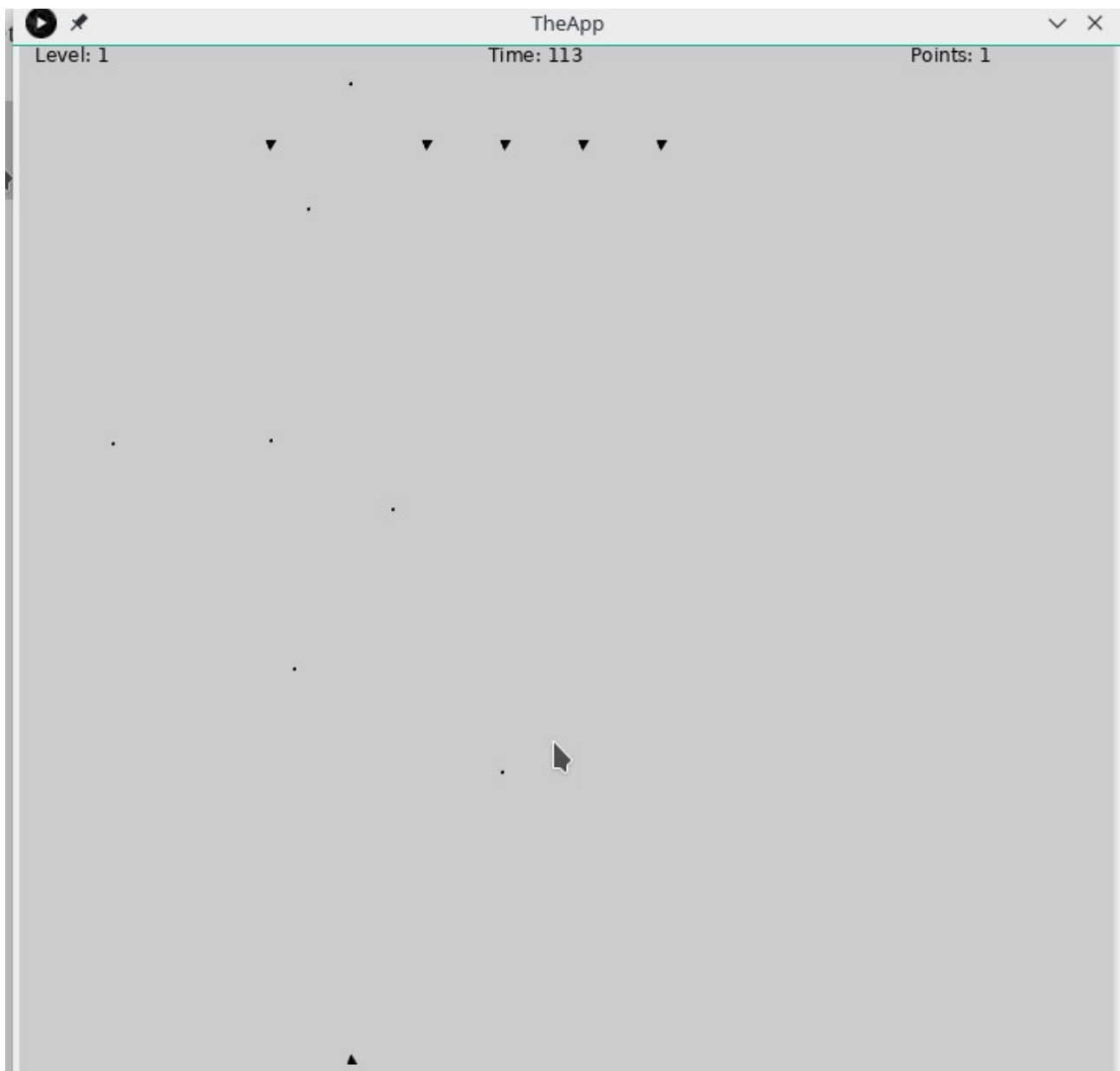
When the player is game over he can restart the game by hitting return.

Gameplay:

Starting the game shows the controls and waits for the user to press enter



Pausing the game stops it and shows a blinking pause string



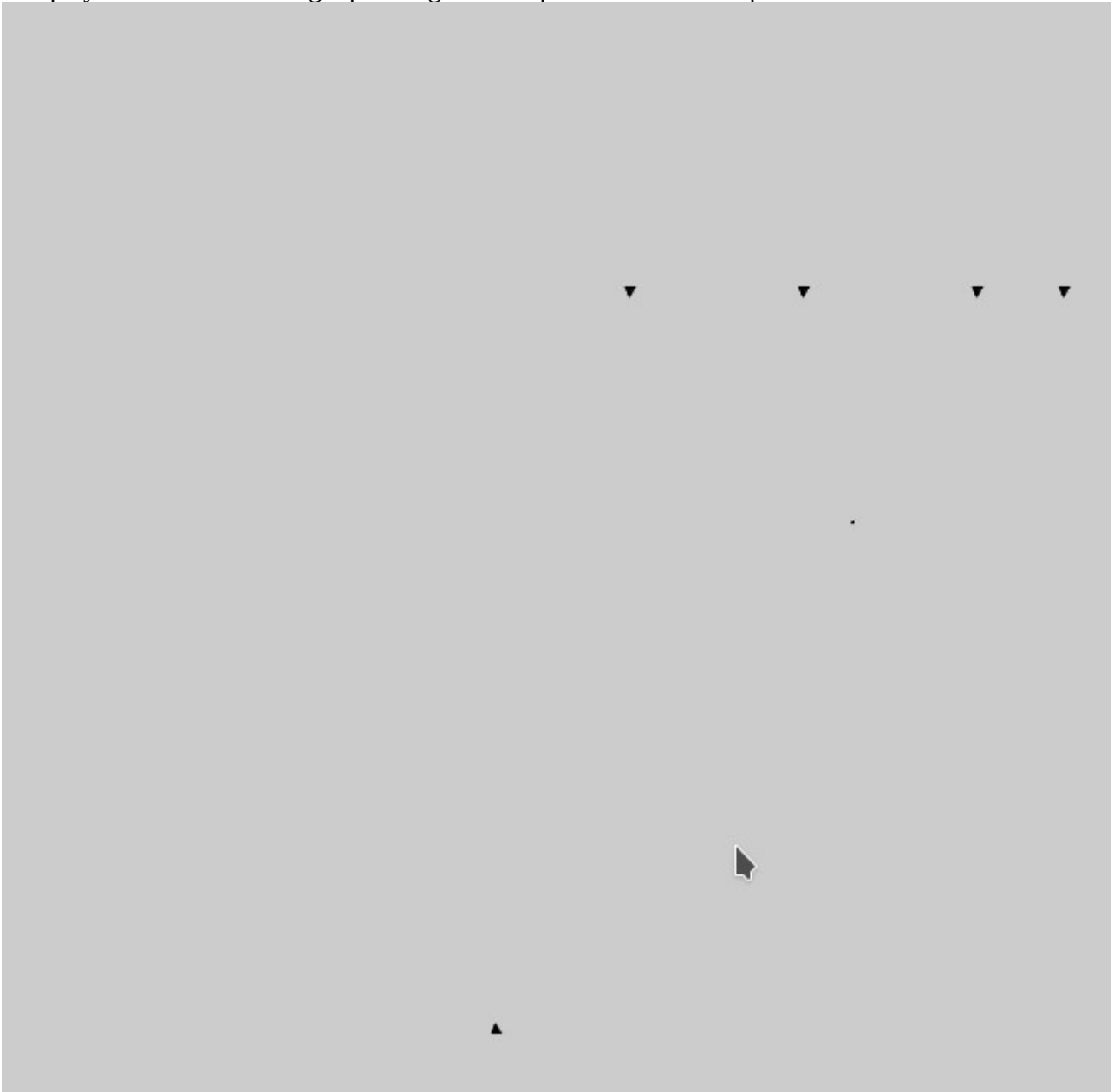
In the upper left corner you can see the current level.

In the top middle the elapsed time is counted.

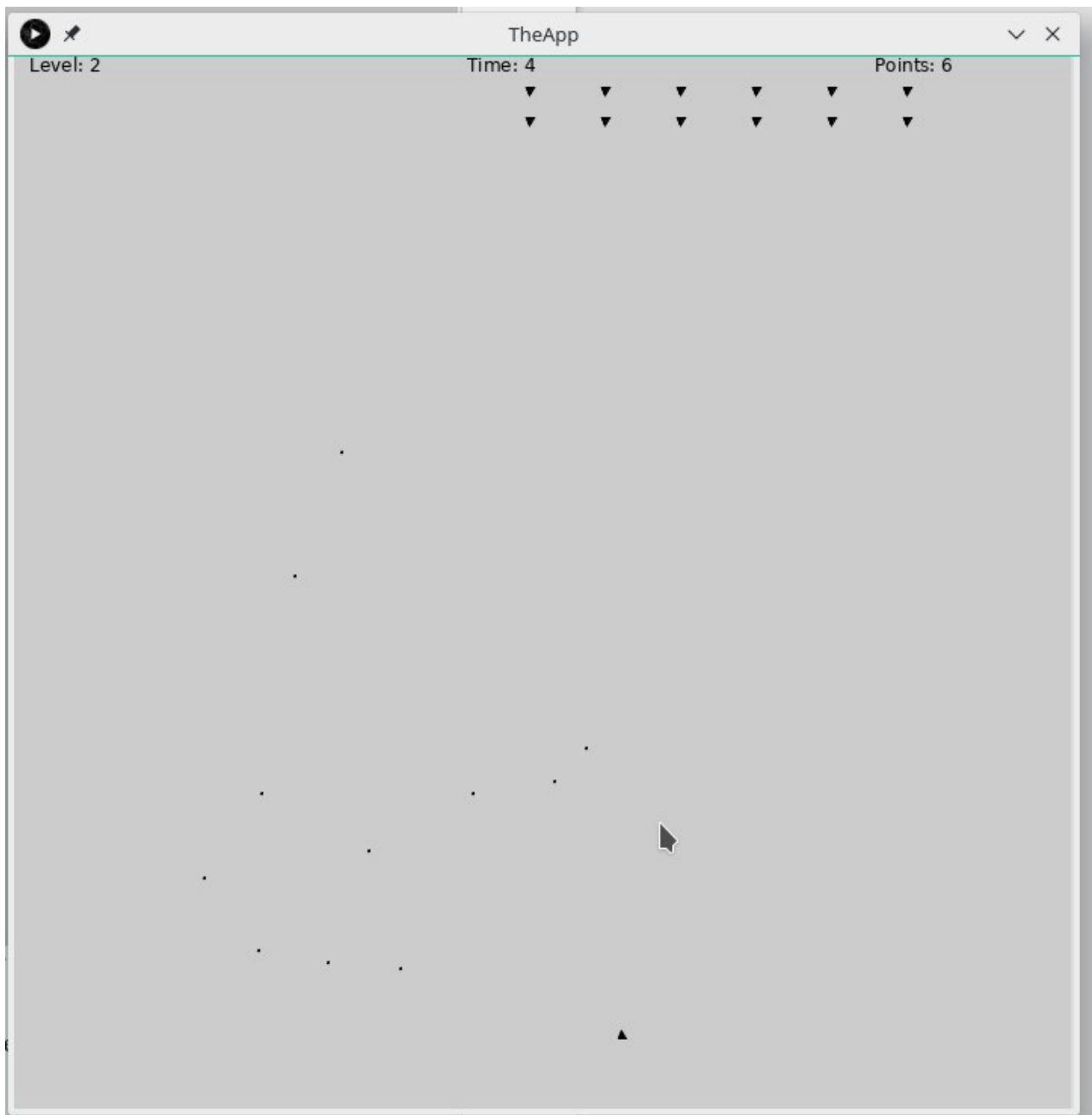
The right upper corner contains the points (1 point for every destroyed invader).

The points are the shot bullets which are moving straight from top to bottom (when the invaders shoot) or from bottom to top (when the player shoots).

The player is drawn as triangle pointing to the top and the invaders point to the bottom.



During the game the invaders fall 50 points (a whole row) and lower the distance to the player until they reach him at the bottom.



When the player reaches the next level the invaders respawn in doubled number.