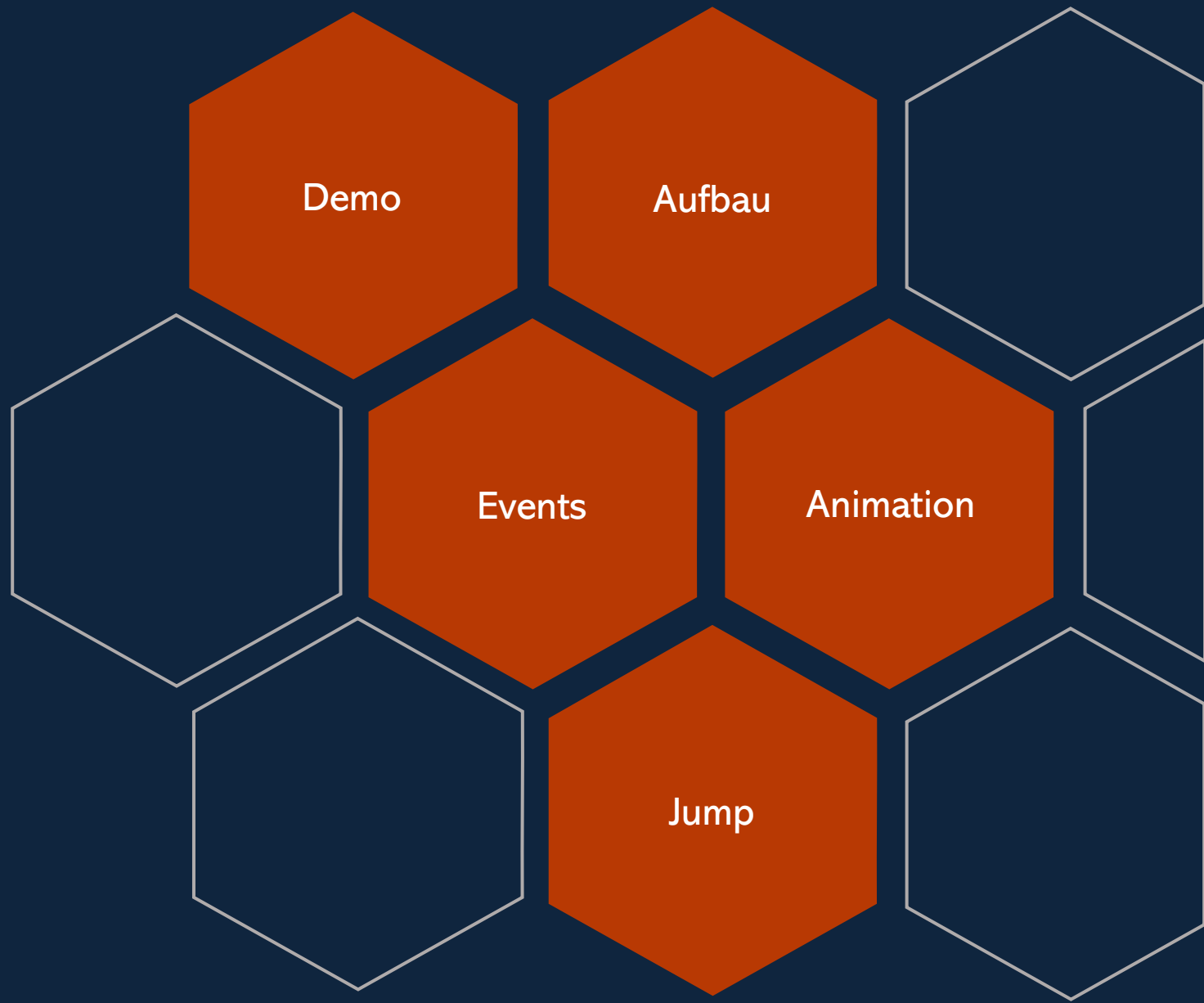
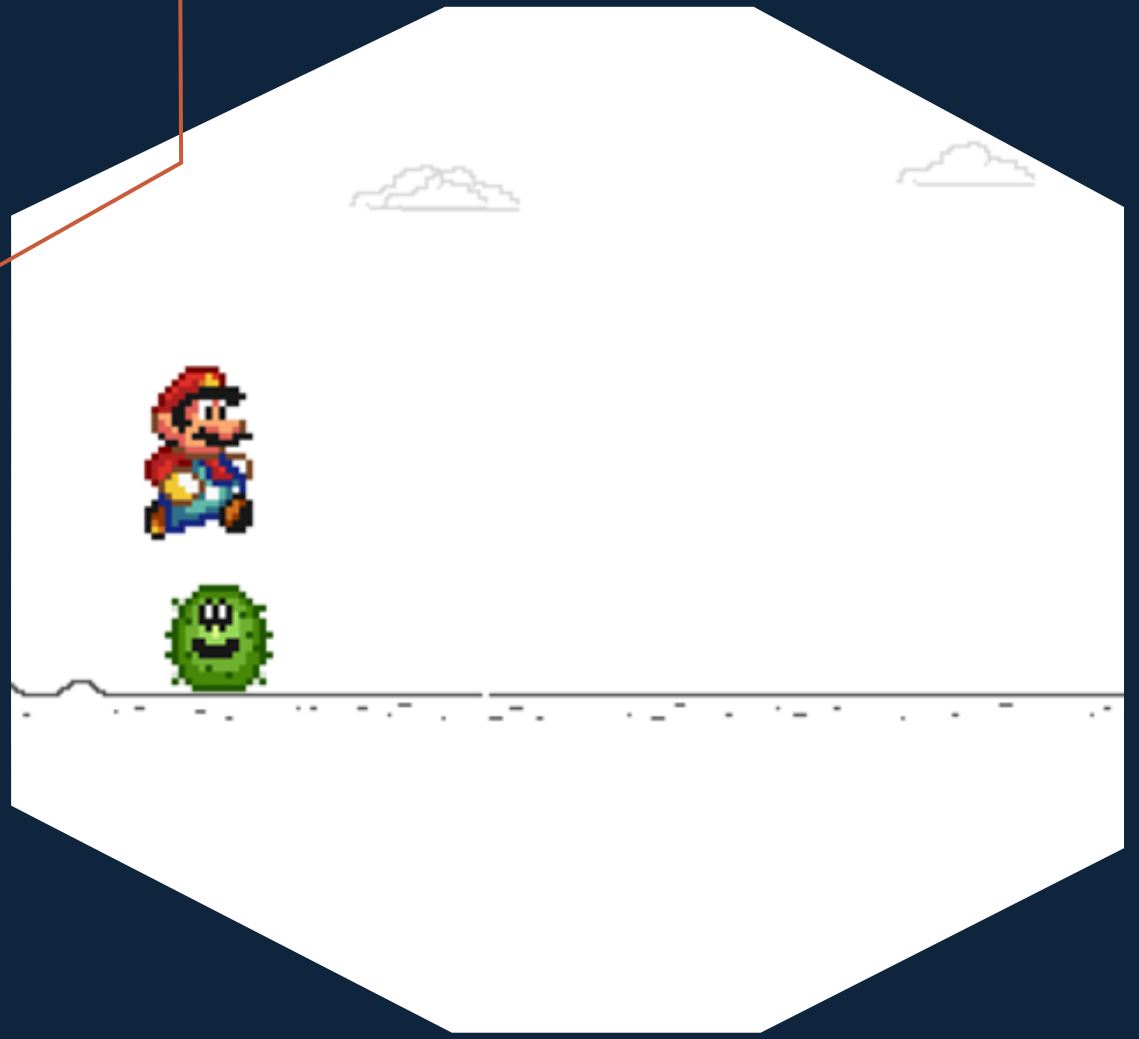
Jump-and-Run-Spiel

Aidan, Lisa, Johanna

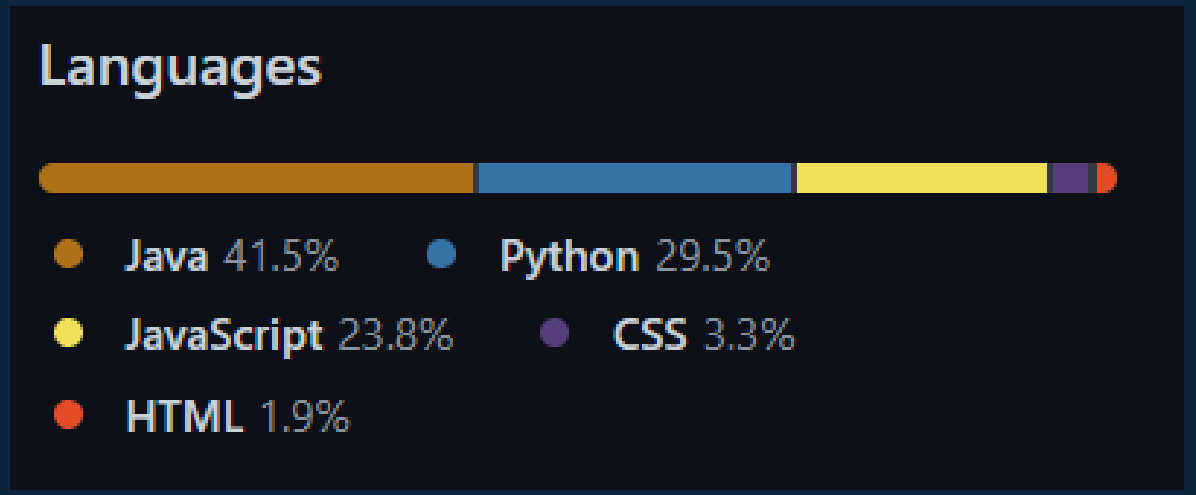
Agenda



Demo



Aufbau



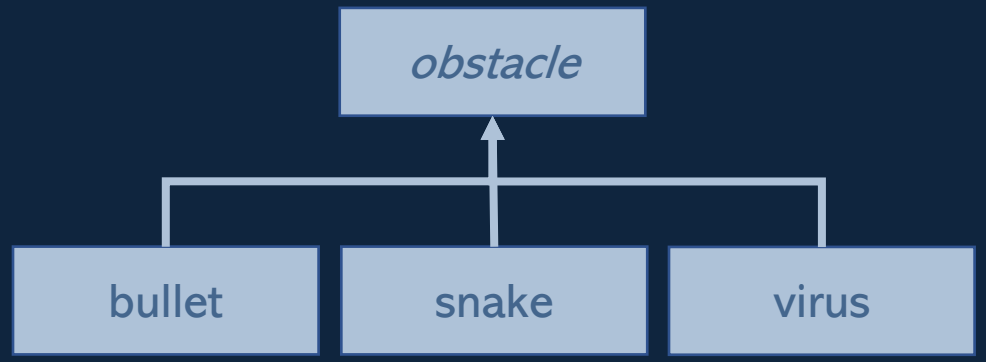
Aufbau - Python

Funktionen

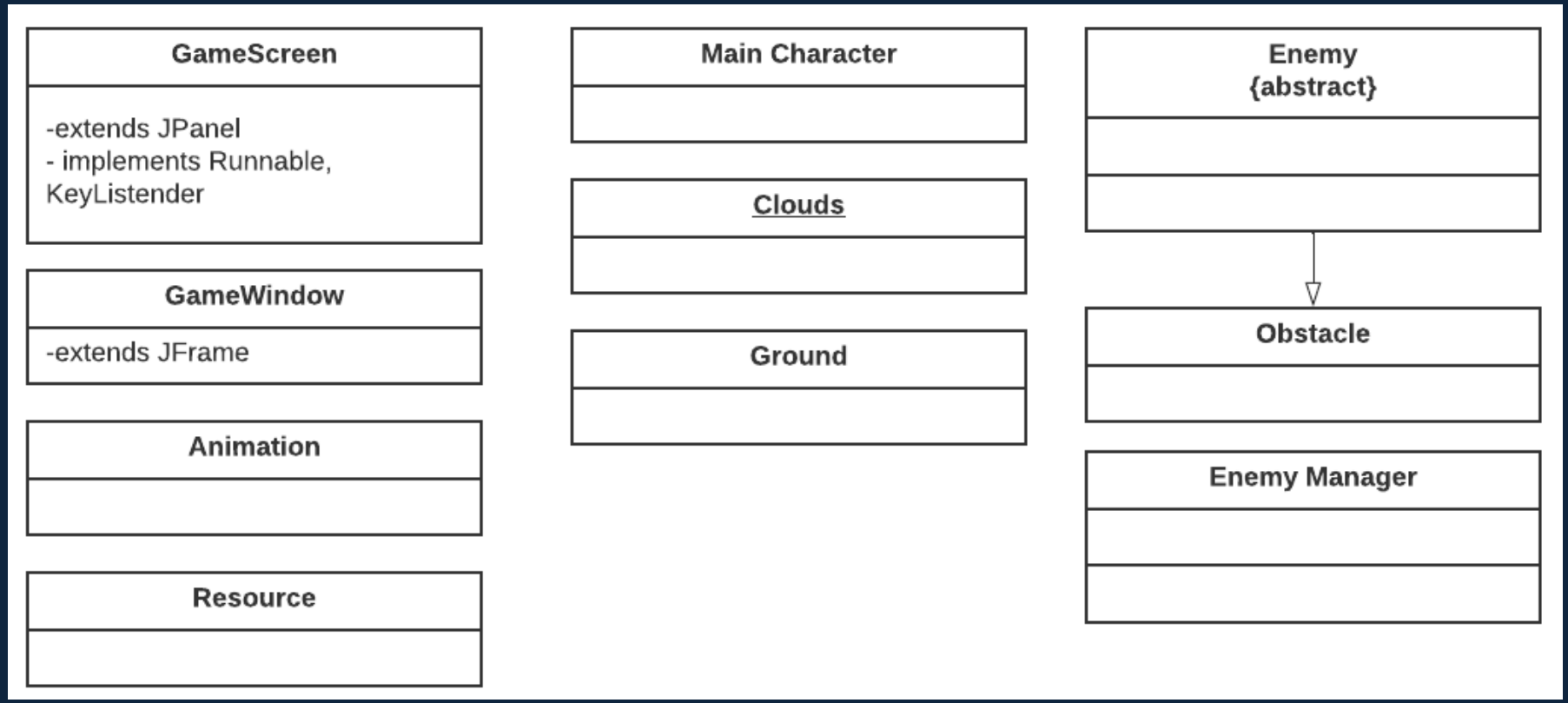
- main
- menu
- highscore

Klassen

- mario
- cloud
- ground



Aufbau - Java


GameScreen
-extends JPanel
- implements Runnable, KeyListender
GameWindow
-extends JFrame
Animation
Resource
Main Character
Clouds
Ground
Enemy
{abstract}
Obstacle
Enemy Manager

Aufbau - Javascript

HTML Klassen
- world
- score
- highScore
- start-screen
- ground
- player

script.js

Hauptfile mit allen Funktionalitäten (Start, Niederlage, Score etc.)

updateCustomProperty.js

Allgemeine Getter und Setter für alle HTML Objekte

player.js

Spielerbewegung mit springen und ducken

ground.js

Bewegung des Bodens

enemy.js

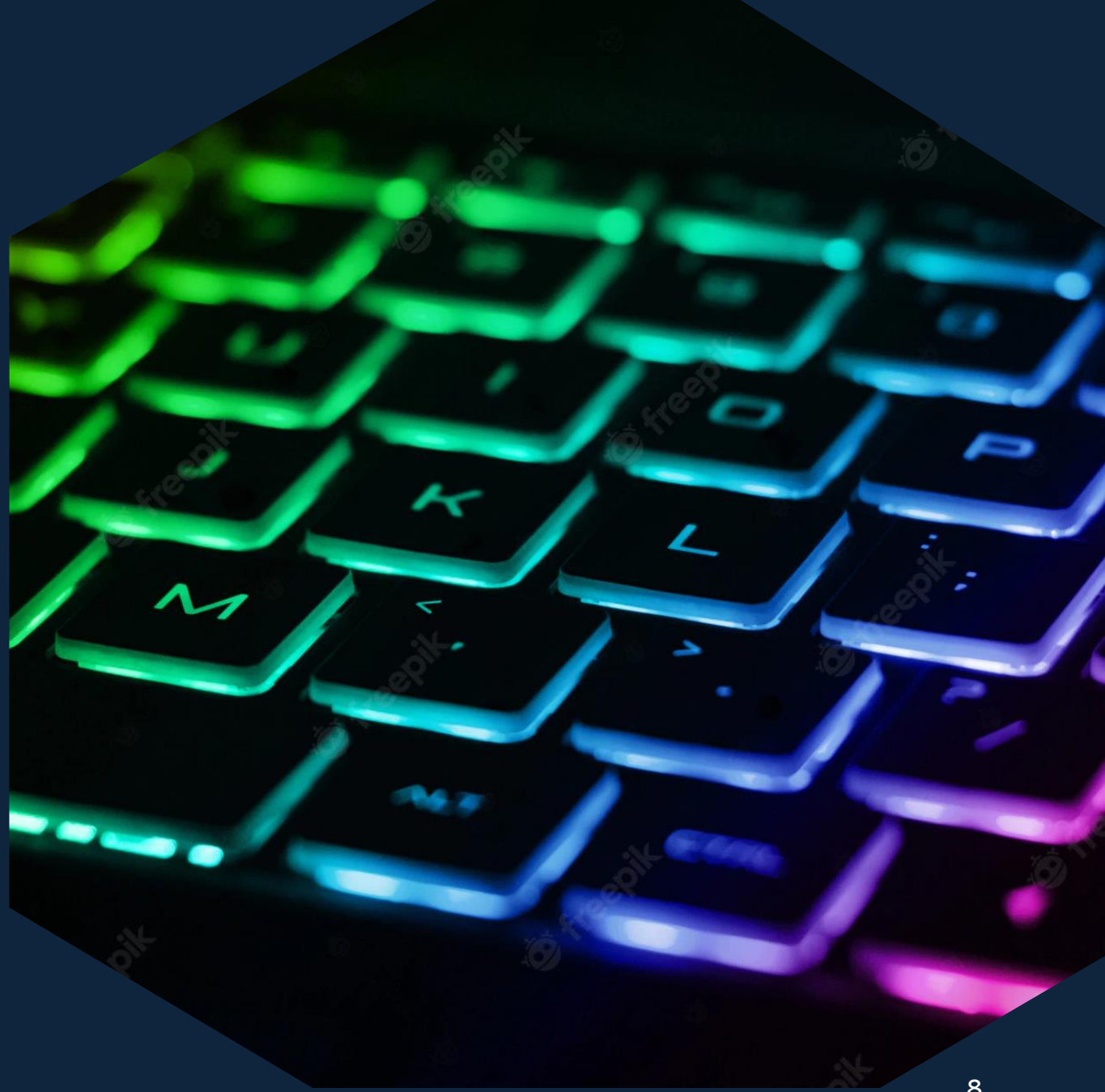
Erstellung und Bewegung verschiedener Gegner

clouds.js

Erstellung und Bewegung der Wolken

Position und Größe wird gesetzt und von JS verändert

Events



Events

Javascript

```javascript
document.addEventListener(
  "keydown",
  handleStart,
  { once: true }
)
```

Java

```java
@Override
public void keyPressed(KeyEvent e) {
    if(gameState == GAME_PLAY_STATE) {
        mainCharacter.jump();
        jumpedInGame = true;
    }
}

@Override
public void keyReleased(KeyEvent e) {
    switch (e.getKeyCode()){
        case KeyEvent.VK_SPACE:
            if(gameState == GAME_BEGIN_STATE){
                gameState = GAME_PLAY_STATE;
            }else if(gameState == GAME_OVER_STATE && jumpedInGame){
                jumpedInGame = false;
            }else if(gameState == GAME_OVER_STATE){
                try {
                    Thread.sleep((long) (GAMEOVER_FRAMES*FRAMETIME));
                }catch(InterruptedException ex){
                    ex.printStackTrace();
                }
                resetGame();
                gameState = GAME_PLAY_STATE;
            }
            break;
    }
}
```

Python

```python
def events(highscore: int):
    from main import main
    for event in pygame.event.get():
        if event.type == pygame.QUIT or event.type == pygame.K_ESCAPE:
            run = False
            exit()
        if event.type == pygame.KEYDOWN:
            main(highscore)
```

Java

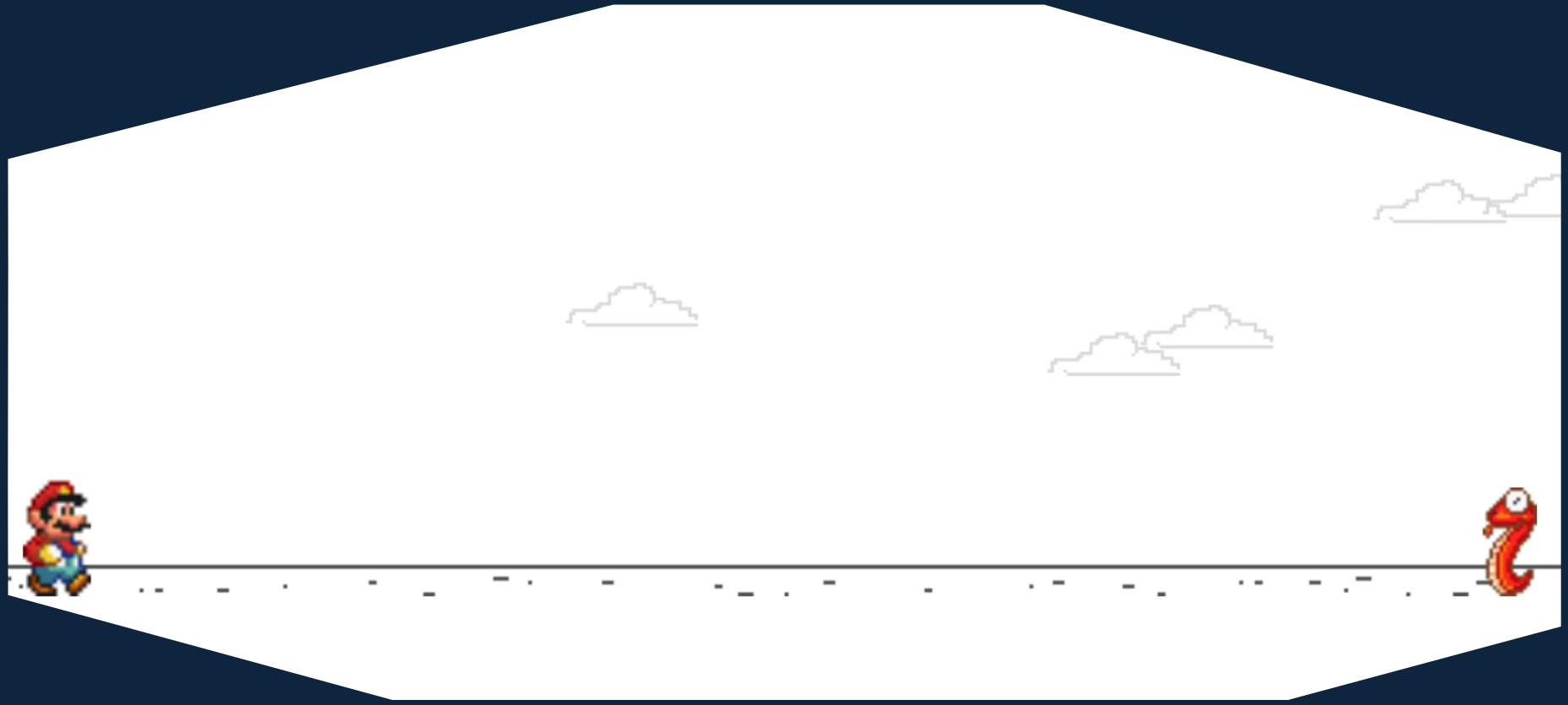
```java
@Override
public void keyPressed(KeyEvent e) {
    if(gameState == GAME_PLAY_STATE) {
        mainCharacter.jump();
        jumpedInGame = true;
    }
}

@Override
public void keyReleased(KeyEvent e) {
    switch (e.getKeyCode()){
        case KeyEvent.VK_SPACE:
            if(gameState == GAME_BEGIN_STATE){
                gameState = GAME_PLAY_STATE;
            }else if(gameState == GAME_OVER_STATE && jumpedInGame){
                jumpedInGame = false;
            }else if(gameState == GAME_OVER_STATE){
                try {
                    Thread.sleep((long) (GAMEOVER_FRAMES*FRAMETIME));
                }catch(InterruptedException ex){
                    ex.printStackTrace();
                }
                resetGame();
                gameState = GAME_PLAY_STATE;
            }
            break;
    }
}
```

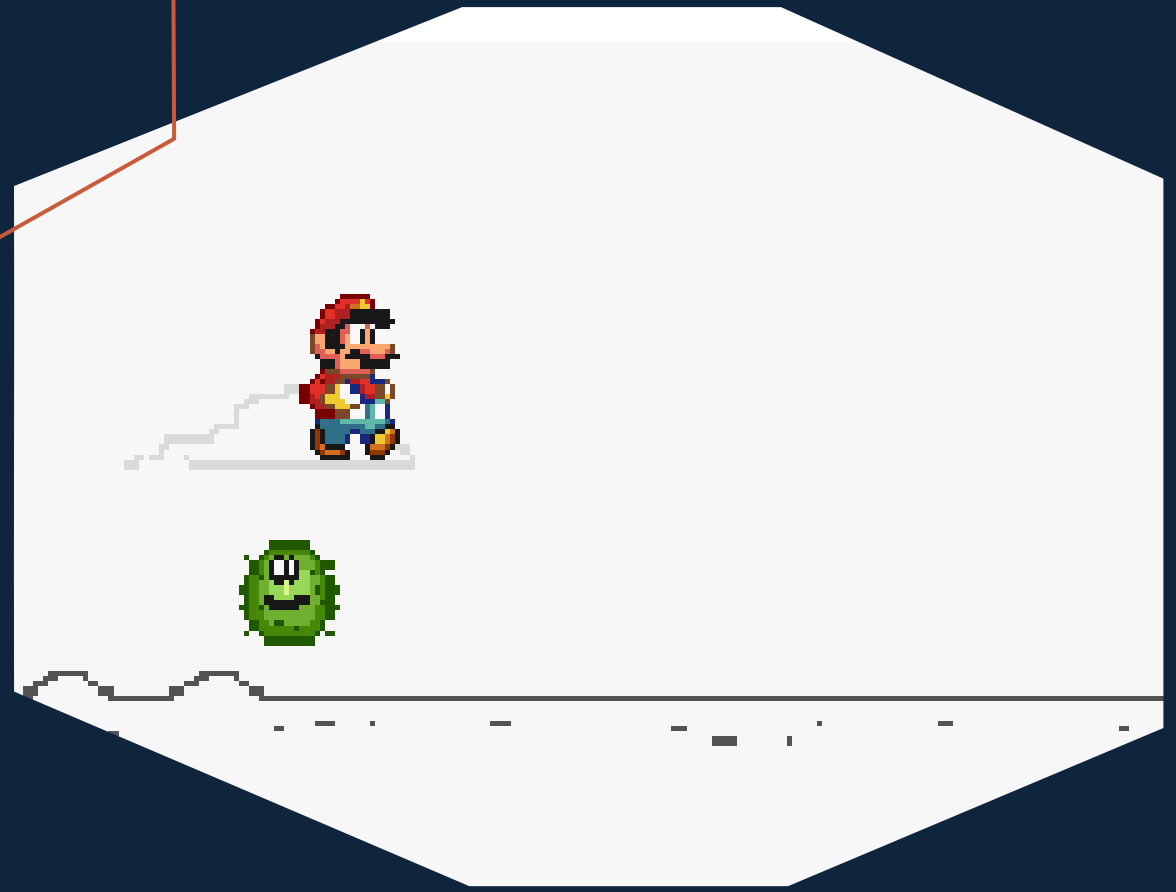
Python

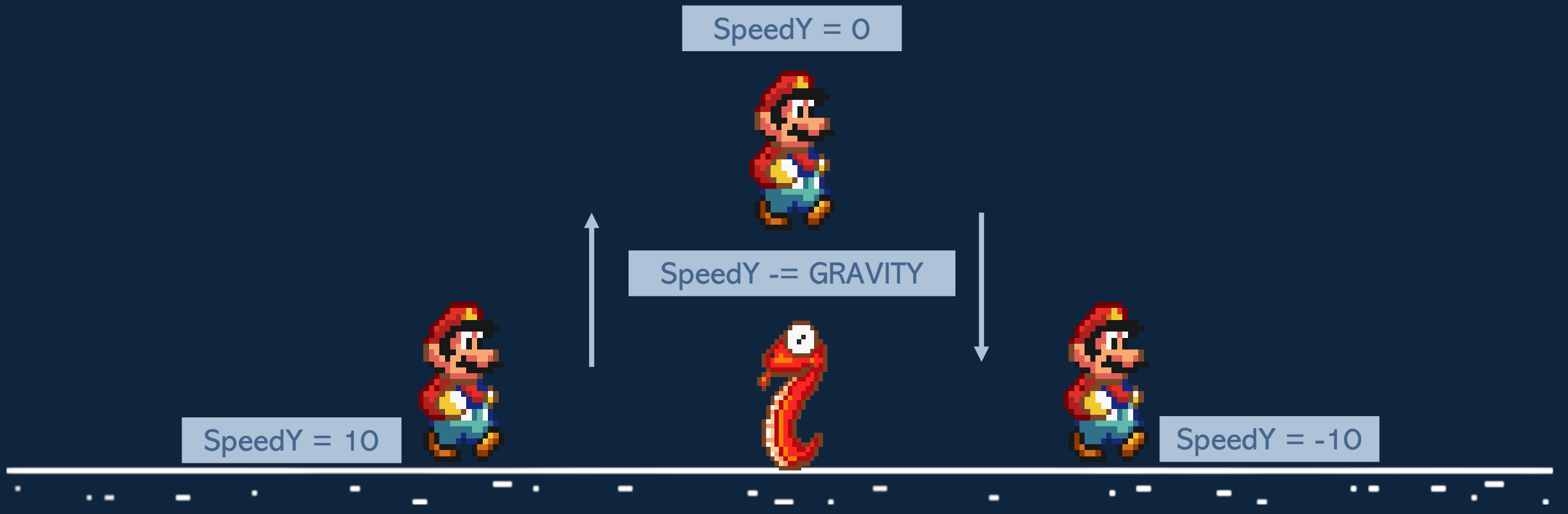
```python
def events(highscore: int):
    from main import main
    for event in pygame.event.get():
        if event.type == pygame.QUIT or event.type == pygame.K_ESCAPE:
            run = False
            exit()
        if event.type == pygame.KEYDOWN:
            main(highscore)
```

Animation



Jump




SpeedY = 0
SpeedY -= GRAVITY
SpeedY = 10
SpeedY = -10

Vielen Dank!