Klasse Button:

void \_\_init\_\_(int, int, int, int, pygame.Image)

boolean check\_if\_clicked(int, int)

pygame.Rect get\_rect()

void setBorder(boolean)

void draw(pygame.Window)

Klasse Chunk:

void \_\_init\_\_(int, Obstacle[])

Obstacle[] get\_obstacles()

Klasse Controller:

void \_\_init\_\_(int, int, int, int, int, int)

void run()

Klasse DeathPanel(Panel):

void \_\_init\_\_(pygame.Window, int())

void draw()

void set\_score(int, boolean)

Klasse Entity:

pygame.Rect hitbox

pygame.Rect coords

void \_\_init\_\_(int(), int())

void move(int, int)

boolean check\_collision(pygame.Rect)

Klasse GamePanel(Panel):

void \_\_init\_\_(pygame.Window, int(), int, int, Model)

void repaint\_background()

void repaint\_player()

void repaint\_obstacles()

pygame.Rect reposition(pygame.Rect, boolean)

void draw()

Klasse Model:

int x

int width

Player player

boolean alive

boolean start

pygame.Image background

pygame.Image start\_screen

void \_\_init\_\_(int, int, int)

void add\_observer(ModelBeobachter)

void remove\_observer(ModelBeobachter)

void update\_observers()

int() get\_dimension()

boolean left\_key()

boolean right\_key()

void space\_key()

Obstacle[] get\_obstacles\_view()

void update\_game()

void onDeath()

void restart\_game()

void load\_data()

boolean updateHighscore()

void save\_data()

Klasse ModelBeobachter:

abstract void update()

Klasse Obstacle(Entity):

boolean death

void \_\_init\_\_(int(), String, boolean)

Klasse Panel:

abstract void draw()

Klasse Player(Entity):

pygame.Image current\_animation

void \_\_init\_\_(int(), int(), int, int)

void space()

void left()

void right()

void update\_current\_animation()

void physics()

(int, boolean) adjust\_collision\_values(int, int, int, Obstacle[])

(int, boolean) update(int, int, int, Obstacle[])

void load\_animation\_img()

Klasse StartPanel(Panel):

Button startButton

Button exitButton

int selected

void \_\_init\_\_(pygame.Window, int(), pygame.Image)

void draw()

void on\_keypress(int)

void select(int)

Klasse View(ModelBeobachter):

Panel panel

void \_\_init\_\_(Model, int, int)

void repaint()

void update()

void update\_start\_screen(int)

void change\_panel(int)

Klasse World:

void \_\_init\_\_()

void reset()

void add\_chunk()

void update(int)

Obstacle[] get\_current\_obstacles(int, int)

Obstacle[] get\_current\_obstacles\_view()

void load\_floating\_objects()

void load\_ground\_objects()

Obstacle[] generate\_objects()