

# “pygame” am Beispiel von “screensaw.py”

Marek Kubica

22. November 2007



# Was soll das sein?

- Python-Binding für SDL
- Simple DirectMedia Layer
- Von Loki (Loki Games) entwickelt
- Einfache (aber auch primitive) API zur Programmierung von Spielen



# “screensaw.py” als Pygame-Anwendung

## Der Ursprung

- 27. Februar 2005 ins Repository eingefügt
- Als eine Art Python-Klon von XScreenSaver gedacht (siehe schlechtes Wortspiel im Namen)
- Seitdem sporadisch erweitert

## Der Umfang von “screensaw.py”

- einige “Bildschirmschoner”, also einfache Demos
- eine Bibliothek die Zufallswerte mit bestimmten Wahrscheinlichkeiten ausgibt
- eine stark optimierte Bibliothek zum Errechnen von Primzahlen (in Python, also keine Wunder zu erwarten)



Vorweg: Ja, es gibt Screenshots.

## Simple Demos

- Fade: Überblendung von schwarz nach weiß und zurück. Der Trenner sozusagen
- Critter: Zwei zufällige Punkte mit weißer Linie verbinden
- Ant: eine Art "Ameise" läuft über den Bildschirm
- Popsquares: blinkende blaue Kästen
- Cube: Ein Gamecube-ähnliches Logo aus Sechsecken konstruiert
- Freefall: Freier Fall



## Interessantere Demos

- Sine Wave: Sinuswelle, mit farbigen Extrema
- Primes: Visualisiert Primzahlen
- AXE Alaska: Drehendes Sechseck. Nicht ganz fertig geworden, keine Motivation mehr
- Textraise: Fliegende Schrifteffekte

## Angefangene, aber nie beendete Demos

- Laser: ebenfalls von XScreenSaver kopiert, konstante Rotationsgeschwindigkeit fertig
- Grail: Bau des Grail-Logos, war aber zu zeitaufwendig
- Magnets: Magnet-Physik-Simulation.



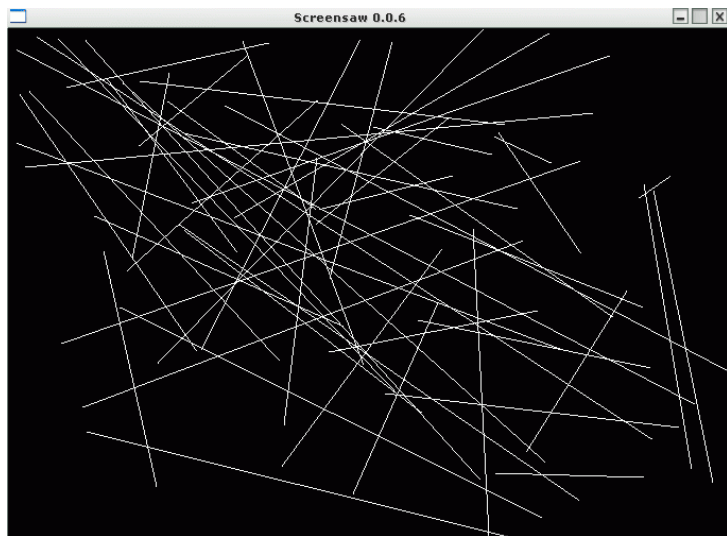


Figure: Computergenerierte Kritzeleien



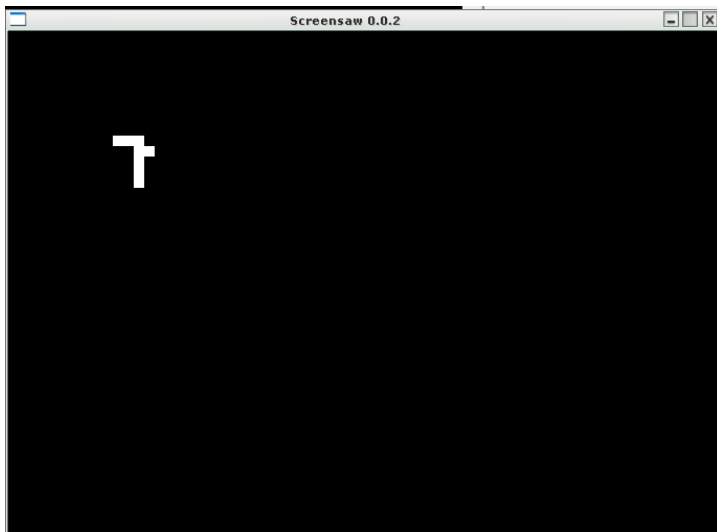


Figure: Ein Quarder, der herumläuft



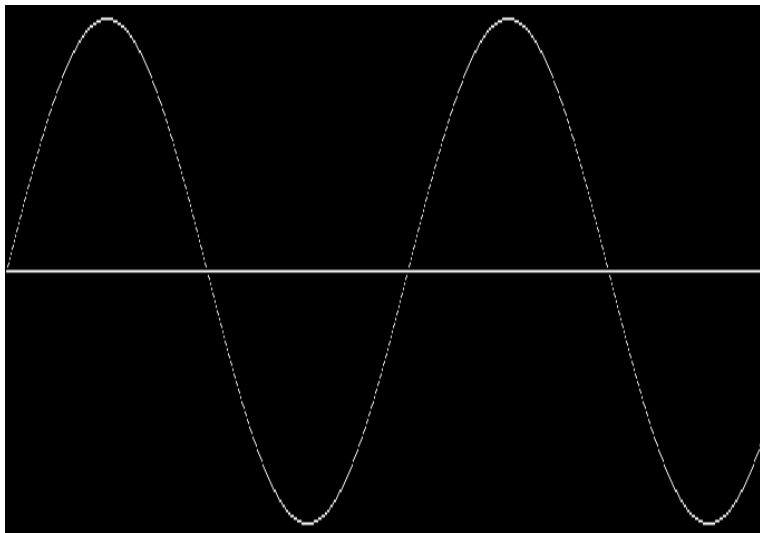


Figure: Sinuswelle





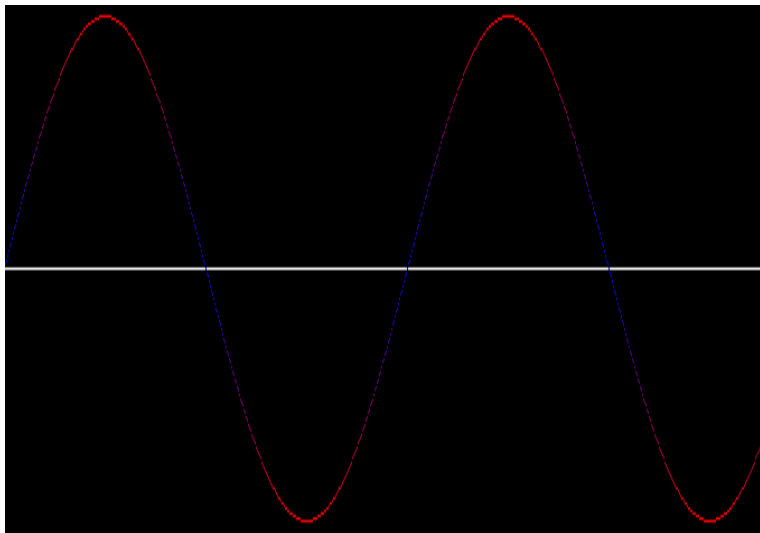


Figure: Sinuswelle mit Extrema



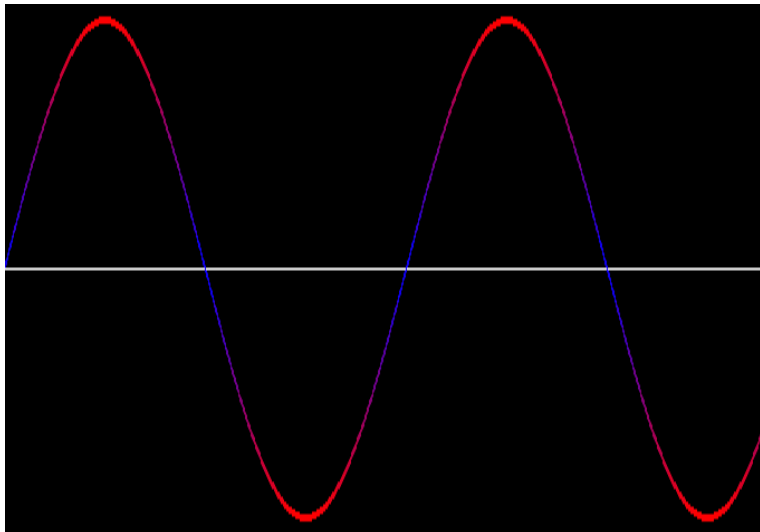


Figure: Farbige Sinuswelle, nachgezogen



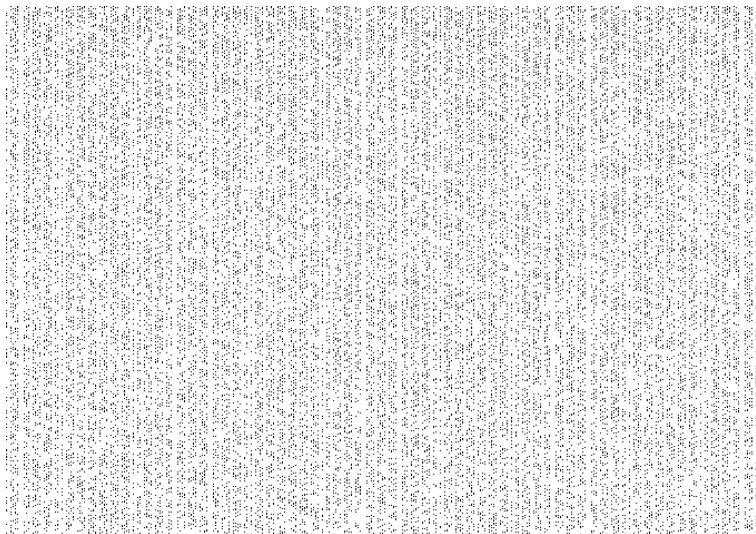


Figure: Primzahlenverteilung, schwarz auf weiß



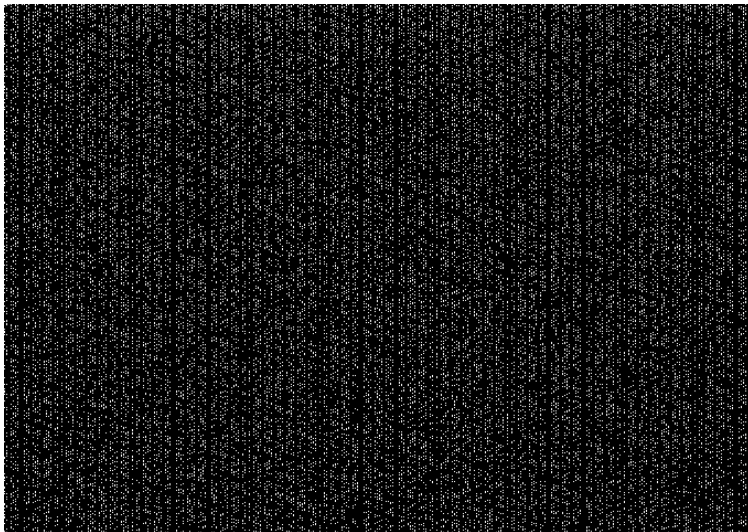


Figure: Primzahlenverteilung, weiß auf schwarz



# Outtakes

Einige Beispiele von des rotierenden Sechsecks, die Probleme mit "pygame" aufzeigen.



Figure: Nach unten bewegt

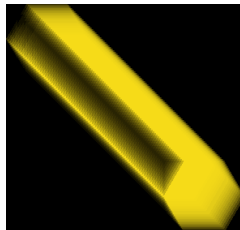


Figure: Seitwärtsbewegung

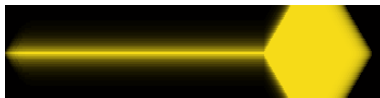


Figure: Nach rechts bewegt



# Probleme mit dem Rotieren

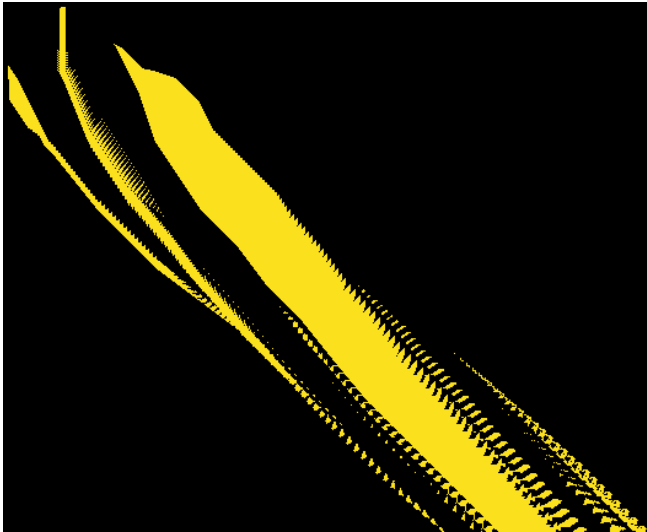


Figure: Das Sechseck, einige Male hintereinander rotiert



## Was sind Sprites?

- Sprites sind Surfaces + ihre Rectangles
- Simpler: Bewegliche Teile auf dem Bildschirm, etwa Raumschiffe, Gegner, Spielfigur.
- Grob: alles was nicht der Hintergrund ist
- Vorteil: Man kann diese Sprites einfacher herumbewegen
- Beispiel: `sprite-demo.py`
- Essentielles Tutorial dazu: <http://kai.vm.bytemark.co.uk/~piman/writing/sprite-tutorial.shtml>



## Problem

Die Ausgabe der Primzahlen war zu langsam. Das lag daran, dass das gesamte Surface ge-updated wurde, immer wenn eine neue Primzahl eingezeichnet wurde.

## Lösung: "Dirty Rectangles"

- Jede Zeichenaktion gibt sogenannte "dirty rectangles" zurück, das sind die Rechtecke die verändert wurden.
- Man kann `pygame.display.update` die zu updatenden "dirty rectangles" mitgeben.
- Voila, nun ist es schneller.





# Pro und Kontra “pygame”

## Es ist nicht *die* Lösung

Also wie man sieht ist es mit “pygame” zwar recht einfach etwas zu bauen, nicht aber trivial. Es gibt auch Schattenseiten.

### Vorteile

- Primitiv, daher einfach
- bietet Grafik, Sound, Film, Input
- Dokumentiert (API, Tutorials)
- Community
- Überall verfügbar (Linux, BSD, Mac, Windows)

### Nachteile

- Entwicklung eingeschlafen (analog zu SDL)
- nur 2D (3D über PyOpenGL)
- direkt *zu* primitiv, zu wenig abstraktion



# Alternativen zu "pygame"

## Direkte Alternativen

- pyglet - sehr neu, aber die Screenshots sehen schon mal klasse aus, Dokumentation ist auch vorhanden

## 3D-Engines

Es gibt viele verschiedene Engines für 3D, aber so wie es aussieht gibt es nicht *die* Engine. Zu den wohl aussichtsreichsten zählen:

- Panda3D - enthält auch eine Physik-Engine
- OGRE 3D - nur Grafik, aber Physik per ODE integrierbar
- Soya3D - eine Engine die speziell für Python geschrieben wurde



# Ende der Präsentation

## pygame selbst

Viele Informationen zu pygame findet man auf der Homepage, <http://www.pygame.org/>. Viele tolle Spiele und Ideen die mit pygame realisiert sind findet man auf <http://www.ludumdare.com> und <http://www.pyweek.org>.

## “screensaw.py” beschaffen

<http://svn.projectdream.org/people/mkubica/snippets/trunk/>

## Danke

Danke fürs Zuhören und ertragen meiner leider nicht so wie geplant gelungenen Folien.

## Fragen? Diskussion?

Gibt es noch Fragen?

