



GRAFIKA GAME

Aditya Wikan Mahastama
mahas@ukdw.ac.id



Tentang Game dan
Representasi Dunia dalam Game

6



TENTANG GAME

- Apa itu game?

SEJARAH GAME

- Ringkasan sejarah game dapat dilihat pada paper berikut oleh Mark Overmars dari Utrecht University - Nederland

http://www.cs.uu.nl/docs/vakken/b2go/docs/history_of_games.pdf

JENIS-JENIS GAME

- Jenis-jenis game (*game genres*) dapat dilihat pada slide berikut oleh Prof. Jason Fritts dari Saint Louis University

http://cs.slu.edu/~fritts/CSCI130_S11/schedule/csci130_game_genres.pdf

REPRESENTASI DUNIA DALAM GAME

- Sebuah game akan memiliki sebuah representasi dunia (world representation) yang menggambarkan keinginan pembuatnya, agar orang yang memainkan game tersebut menangkap esensi dari dunia yang ingin digambarkan tersebut

Elemen representasi dunia dalam game adalah:

- Representasi Peta (Map Representation)
- Representasi Objek

REPRESENTASI DUNIA DALAM GAME

- Mari kita lihat video game engine berikut:

MAP REPRESENTATION

- Berfungsi sebagai lapis koordinat dasar dari sebuah game. Koordinat ini akan digunakan untuk meletakkan objek-objek dalam game, baik itu elemen peta yang statis, maupun objek-objek lain yang dinamis (bergerak)
- Bisa terdiri dari **grid** (bangun serupa berulang) maupun **poligonal** (bangun tidak serupa tidak berulang)
- Satu game bisa memiliki beberapa lapis (*layer*) representasi peta, misal satu untuk dunia, satu untuk karakter, dengan jenis yang berbeda.
- Representasi peta belum tentu sama dengan gerakan yang nampak

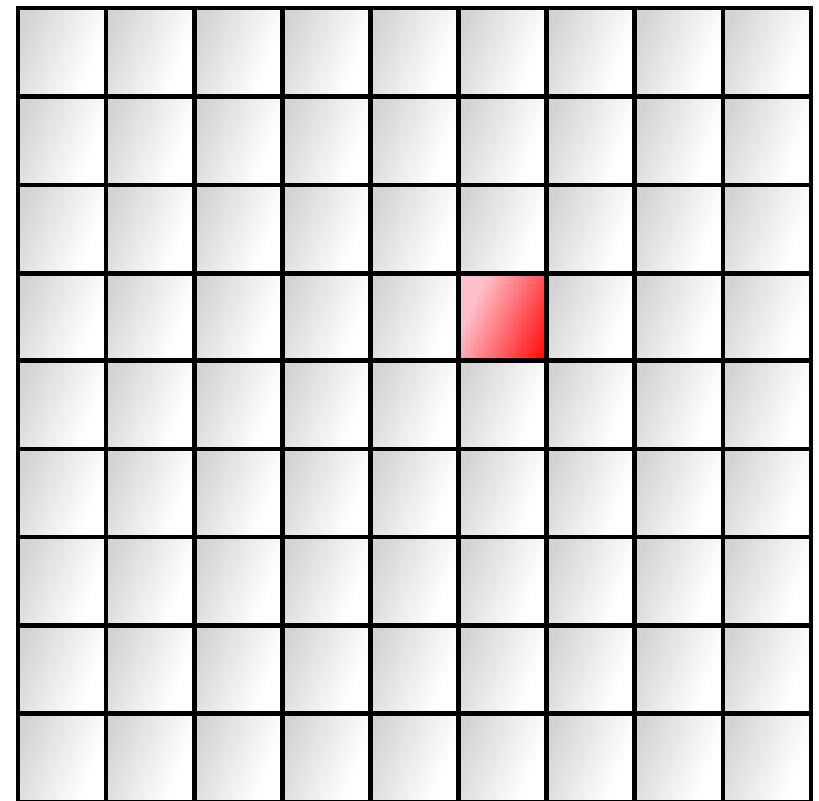
GRID

- Grid adalah membagi papan permainan menjadi petak-petak yang didasarkan pada sebuah/beberapa bangun (datar/ruang), yang berulang dan harus dapat menutup seluruh permukaan papan permainan.
- Untuk dua dimensi, pola grid didasarkan pada *tiling of regular polygons* yang sudah digunakan sebagai pola mosaik sejak masa lampau.
- Pola lengkap dapat dilihat di Wikipedia

BENTUK GRID YANG UMUM DIGUNAKAN

- **Segi Empat**

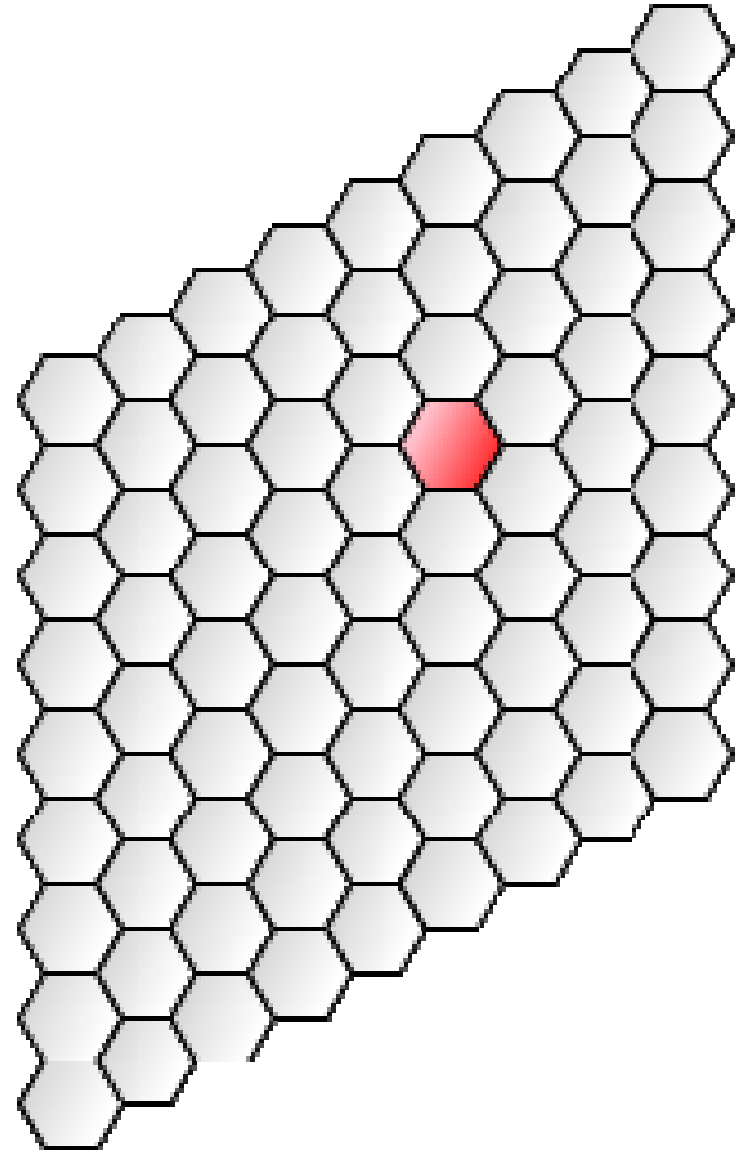
Bentuk grid paling umum adalah segi empat. Sederhana, mudah digunakan dan cocok dengan layar komputer. Lokasi dapat menggunakan sistem koordinat kartesian (x,y) dan sumbunya orthogonal. Sistem koordinatnya akan tetap sama meskipun gridnya diproyeksikan secara isometrik atau aksonometrik.



BENTUK GRID YANG UMUM DIGUNAKAN

- **Heksagon (segi enam)**

Heksagon digunakan di game papan (*board games*), dan beberapa game lain yang mempertimbangkan distorsi jarak yang lebih kecil dari grid segi empat. Ini karena heksagon memiliki tetangga non-diagonal lebih banyak dari pada segi empat. Heksagon menyamarkan kekakuan, dan juga bisa ditemukan di alam.

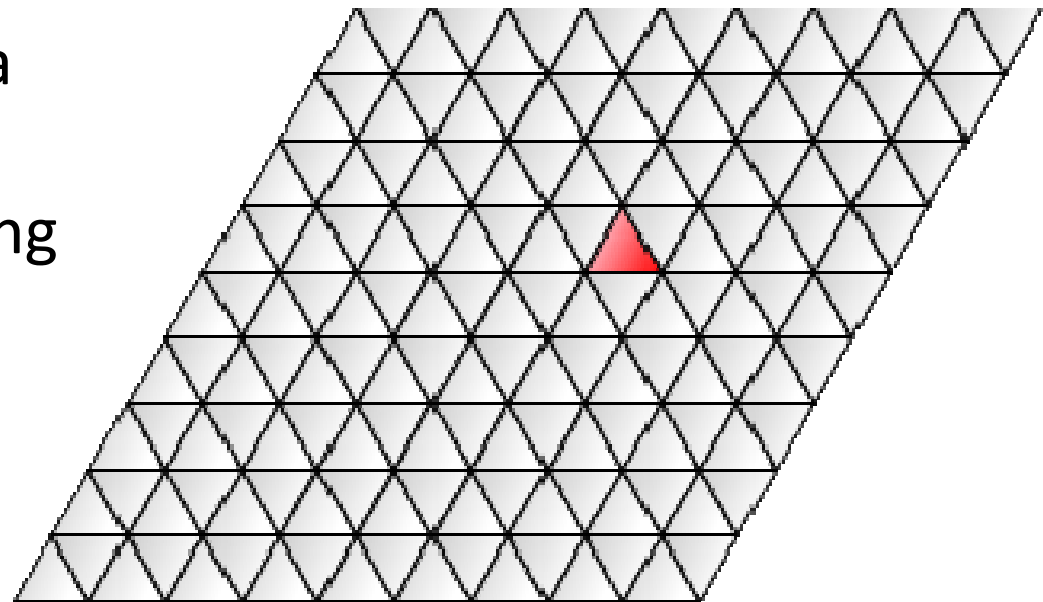


BENTUK GRID YANG UMUM DIGUNAKAN

- **Segi Tiga**

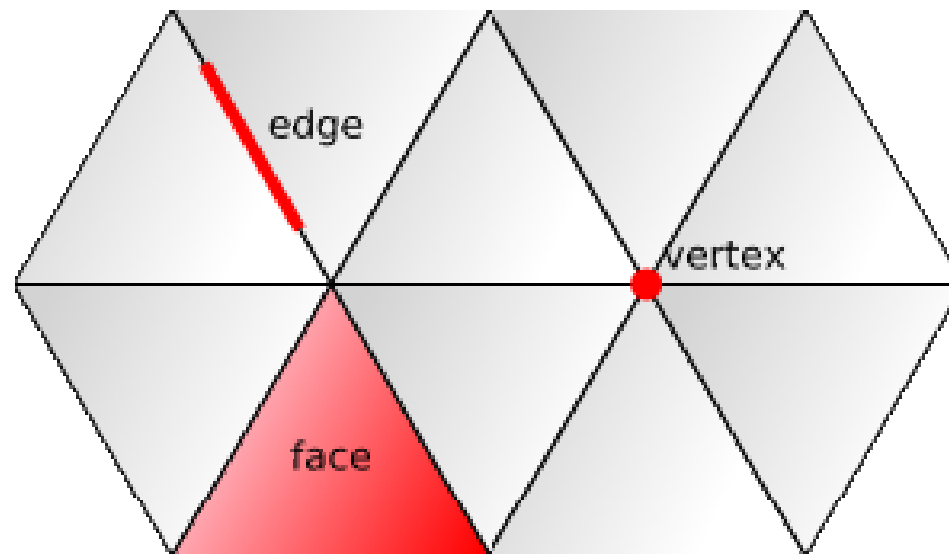
Segi tiga sangat umum di dunia 3D, tetapi jarang digunakan untuk pemetaan game.

Kekurangan dari segi tiga adalah di samping bentuknya yang tidak lazim, kelilingnya besar dengan luasan area yang kecil - bertolak belakang dengan heksagon.



ELEMEN GRID

- Grid memiliki 3 elemen: face (muka/ubin/petak), edge (tepi) dan verteks (titik). Muka adalah permukaan dua dimensi yang dibatasi oleh tepi. Tepi adalah segmen garis satu dimensi yang berujung pada dua titik.

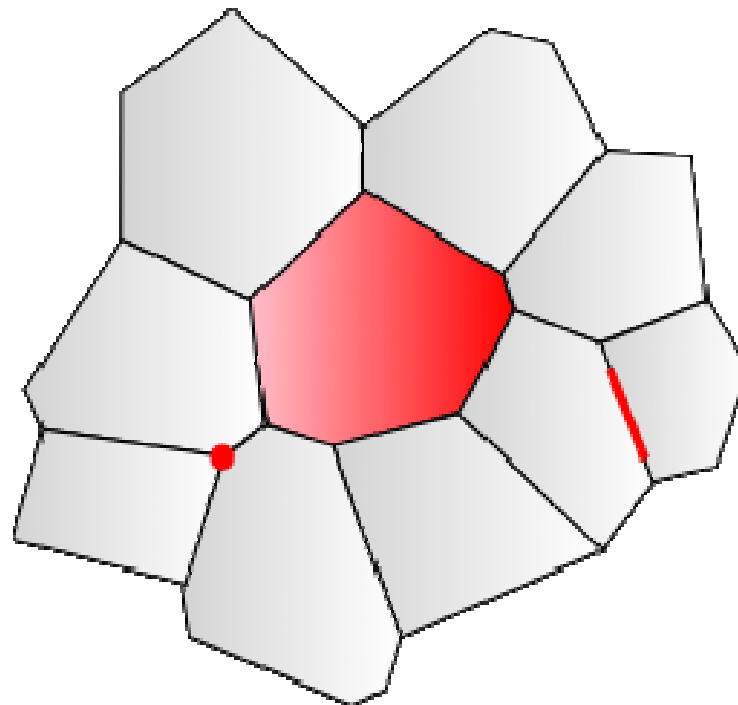


ELEMEN GRID

- Umumnya permainan “barat” seperti catur dan *checkers* menggunakan muka sebagai acuan, sedangkan permainan “timur” seperti *go* dan halma, mengacu pada titiknya. Ada pula permainan lain seperti rolet yang menggunakan ketiga elemen di atas.

BAGAIMANA DENGAN POLIGON?

- Seandainya kita memutuskan untuk membuat grid dari poligon-poligon yang bervariasi, kita tetap dapat menggunakan ketiga elemen tersebut.

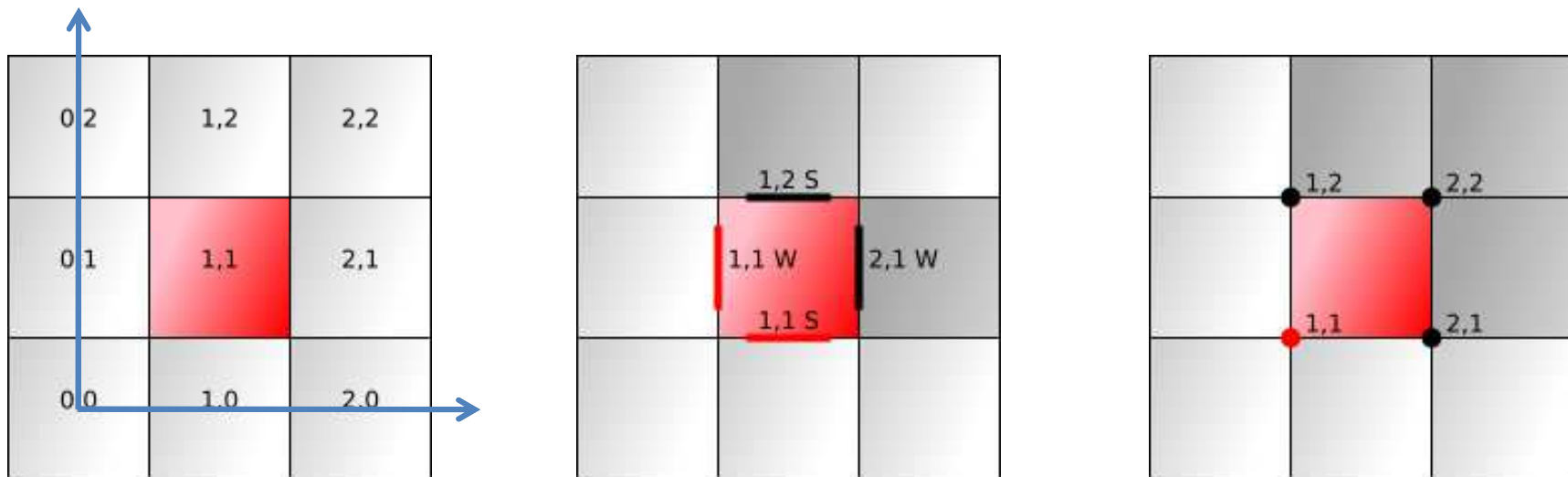


SISTEM KOORDINAT GRID

- Bagian berikut akan membahas mengenai ide pemberian sistem koordinat pada grid untuk membantu memberikan gambaran saat menyusun struktur data yang akan digunakan dalam game.

SISTEM KOORDINAT GRID

- Segi Empat

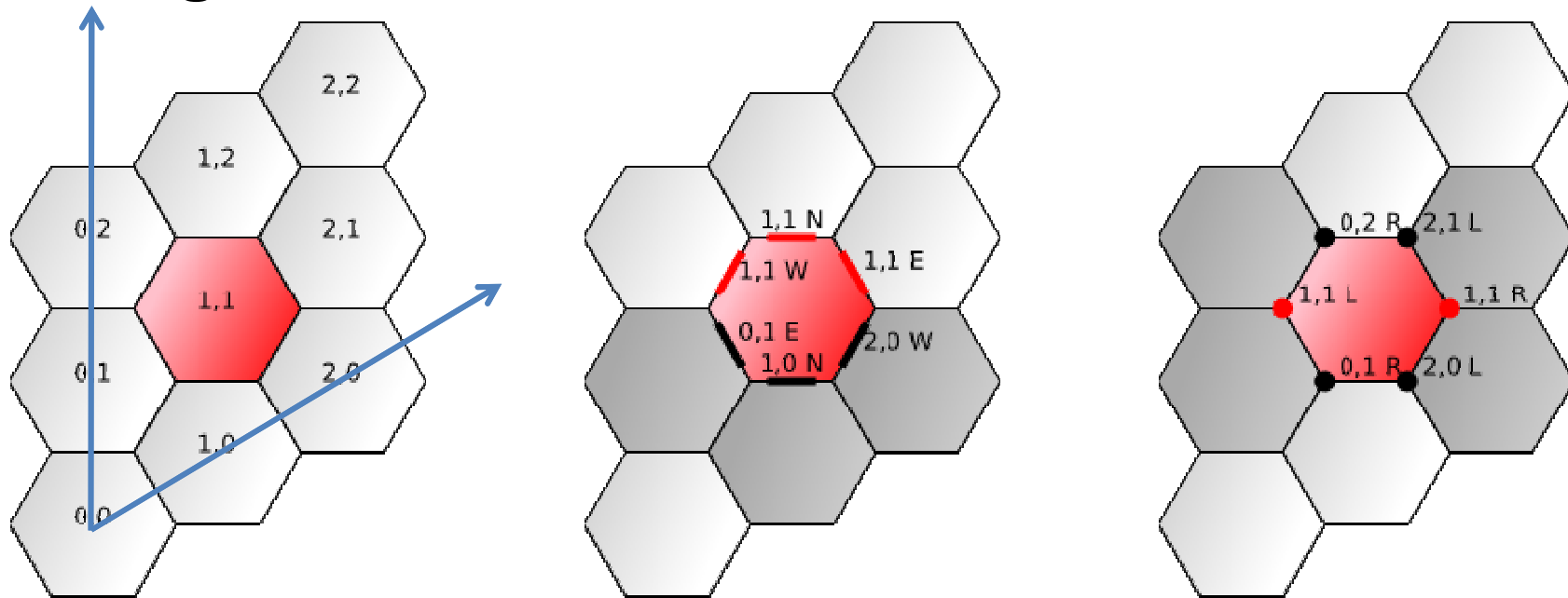


Catatan: contoh di atas menggunakan sumbu Y positif ke arah atas.

Sistem koordinatnya cukup mudah dimengerti, kecuali untuk tepi (edge) kita menggunakan basis muka, dengan penambahan W untuk sisi kiri dan S untuk sisi bawah (baliklah jika sumbu Y positif ke arah bawah)

SISTEM KOORDINAT GRID

- Heksagon

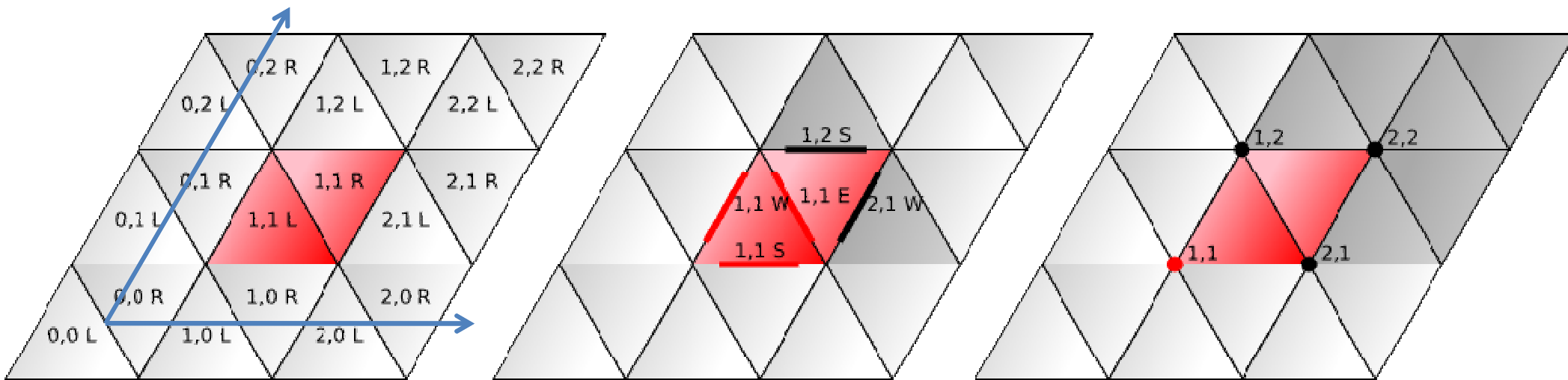


Catatan: contoh di atas menggunakan sumbu Y positif ke arah atas.

Sistem koordinatnya menggunakan kartesian dua dimensi yang dipencengkan untuk **muka**, untuk **tepi** kita menggunakan basis muka, dengan penambahan E, N, W untuk sisi kiri, atas dan kanan (baliklah posisi N jika sumbu Y positif ke arah bawah). Untuk **titik** kita juga menggunakan basis muka, dengan mengambil dua titik saja pada kiri dan kanan muka.

SISTEM KOORDINAT GRID

- Segi Tiga



Catatan: contoh di atas menggunakan sumbu Y positif ke arah atas.

Segi tiga juga menggunakan koordinat kartesian yang dipencengkan dalam bentuk jajaran genjang, kemudian dibagi dua segmen yaitu L (kiri) dan R (kanan) untuk **muka**. Untuk **tepi** basisnya muka dengan sisi W, E, S (silakan dibalik untuk sumbu Y positif ke bawah). Untuk **titik** tidak ada masalah, menggunakan koordinat titik sudut yang mendekati pusat sumbu (0,0)

HUBUNGAN ANTAR ELEMEN GRID

Part A (black)	Part B (red)		
	Face	Edge	Vertex
Face	Neighbors:	Borders:	Corners:
Edge	Joins:	Continues:	Endpoints:
Vertex	Touches:	Protrudes:	Adjacent:

ALGORITMA HUBUNGAN ANTAR ELEMEN GRID

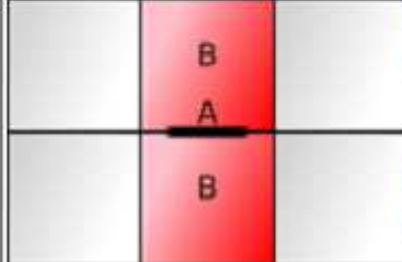
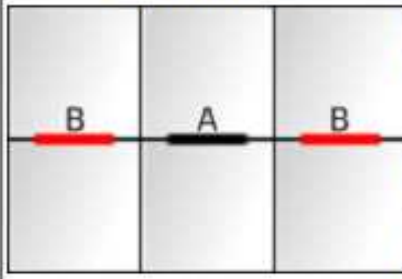
- Berikut ini algoritma untuk menentukan lokasi elemen grid yang berwarna merah, dari lokasi elemen acuan yang berwarna hitam.

Relationship	Shape		
	Square	Hexagon	Triangle
Neighbors:			
	$(u,v) \rightarrow (u,v+1)$ $(u+1,v) (u,v-1),$ $(u-1,v)$	$(u,v) \rightarrow (u,v+1)$ $(u+1,v) (u+1,v-1)$ $(u,v-1) (u-1,v)$ $(u-1,v+1)$	$(u,v,L) \rightarrow (u,v,R)$ $(u,v-1,R) (u-1,v,R)$ $(u,v,R) \rightarrow (u,v+1,L)$ $(u+1,v,L) (u,v,L)$

ALGORITMA HUBUNGAN ANTAR ELEMEN GRID

Relationship	Shape		
	Square	Hexagon	Triangle
Borders: 	$(u,v) \rightarrow$ $(u,v+1,S)$ $(u+1,v,W)$ $(u,v,S) (u,v,W)$	$(u,v) \rightarrow (u,v,N) (u,v,E)$ $(u+1,v-1,W) (u,v-1,N)$ $(u-1,v,E) (u,v,W)$	$(u,v,L) \rightarrow (u,v,E)$ $(u,v,S) (u,v,W)$ $(u,v,R) \rightarrow (u,v+1,S)$ $(u+1,v,W) (u,v,E)$
Corners: 	$(u,v) \rightarrow$ $(u+1,v+1)$ $(u+1,v) (u,v)$ $(u,v+1)$	$(u,v) \rightarrow (u+1,v,L)$ $(u,v,R) (u+1,v-1,L)$ $(u-1,v,R) (u,v,L)$ $(u-1,v+1,R)$	$(u,v,L) \rightarrow (u,v+1)$ $(u+1,v) (u,v)$ $(u,v,R) \rightarrow (u+1,v+1)$ $(u+1,v) (u,v+1)$

ALGORITMA HUBUNGAN ANTAR ELEMEN GRID

Relationship	Shape		
	Square	Hexagon	Triangle
Joins: 	$(u,v,W) \rightarrow (u,v)$ $(u-1,v)$	$(u,v,N) \rightarrow (u,v+1)$ (u,v) $(u,v,E) \rightarrow (u+1,v)$ (u,v)	$(u,v,S) \rightarrow (u,v,L)$ $(u,v-1,R)$ $(u,v,E) \rightarrow (u,v,R)$ (u,v,L)
Continues: 	$(u,v,W) \rightarrow$ $(u,v+1,W)$ $(u,v-1,W)$ $(u,v,S) \rightarrow$ $(u+1,v,S)$ $(u-1,v,S)$	No edges continue straight in a hexagonal grid	$(u,v,W) \rightarrow (u,v+1,W)$ $(u,v-1,W)$ $(u,v,E) \rightarrow (u+1,v-1,E)$ $(u-1,v+1,E)$ $(u,v,S) \rightarrow (u+1,v,S)$ $(u-1,v,S)$

ALGORITMA HUBUNGAN ANTAR ELEMEN GRID

Relationship	Shape		
	Square	Hexagon	Triangle
Endpoints: 	$(u,v,W) \rightarrow (u,v+1) (u,v)$ $(u,v,S) \rightarrow (u+1,v) (u,v)$	$(u,v,N) \rightarrow (u+1,v,L)$ $(u-1,v+1,R)$ $(u,v,E) \rightarrow (u,v,R)$ $(u+1,v,L)$ $(u,v,W) \rightarrow (u-1,v+1,R)$ (u,v,L)	$(u,v,W) \rightarrow (u,v+1) (u,v)$ $(u,v,E) \rightarrow (u+1,v) (u,v+1)$ $(u,v,S) \rightarrow (u+1,v) (u,v)$
Touches: 	$(u,v) \rightarrow (u,v)$ $(u,v-1) (u-1,v-1)$ $(u-1,v)$	$(u,v,L) \rightarrow (u,v) (u-1,v)$ $(u-1,v+1)$ $(u,v,R) \rightarrow (u+1,v)$ $(u+1,v-1) (u,v)$	$(u,v) \rightarrow (u-1,v,R)$ $(u,v,L) (u,v-1,R)$ $(u,v-1,L) (u-1,v-1,R)$ $(u-1,v,L)$

ALGORITMA HUBUNGAN ANTAR ELEMEN GRID

Relationship	Shape		
	Square	Hexagon	Triangle
Protrudes: 	$(u,v) \rightarrow (u,v,W)$ $(u,v,S) (u,v-1,W)$ $(u-1,v,S)$	$(u,v,L) \rightarrow (u,v,W)$ $(u-1,v,E) (u-1,v,N)$ $(u,v,R) \rightarrow (u+1,v-1,N)$ $(u+1,v-1,W) (u,v,E)$	$(u,v) \rightarrow (u,v,W)$ $(u,v,S) (u,v-1,E)$ $(u,v-1,W) (u-1,v,S)$ $(u-1,v,E)$
Adjacent: 	$(u,v) \rightarrow (u,v+1)$ $(u+1,v) (u,v-1)$ $(u-1,v)$	$(u,v,L) \rightarrow (u-1,v+1,R)$ $(u-1,v,R) (u-2,v+1,R)$ $(u,v,R) \rightarrow (u+2,v-1,L)$ $(u+1,v-1,L) (u+1,v,L)$	$(u,v) \rightarrow (u,v+1)$ $(u+1,v) (u+1,v-1)$ $(u,v-1) (u-1,v)$ $(u-1,v+1)$

MENYIMPAN RELASI ANTAR ELEMEN GRID BERBEDA

- Untuk mempermudah, kita dapat menyimpan relasi antar elemen yang berbeda pada petak yang sama dengan mendefinisikannya melalui tabel pada basis data. Contoh berikut untuk grid berbentuk segi empat:

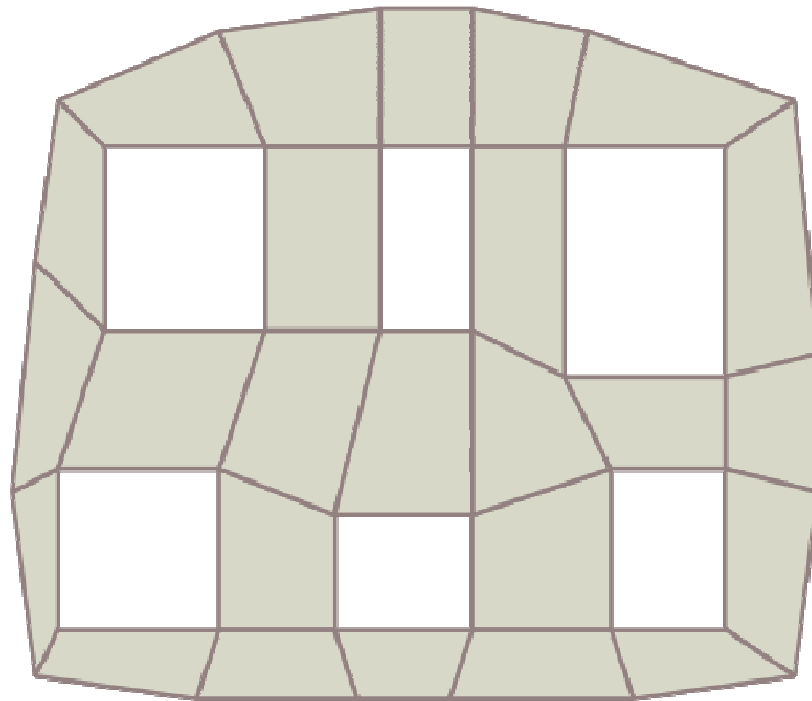
Face	Edge
0,0	0,0,S
0,0	0,0,W
0,0	0,1,S
0,0	1,0,W
1,0	1,0,S
1,0	1,0,W
1,0	1,1,S
1,0	2,0,W

POLIGONAL

- Peta poligonal membagi papan permainan menjadi segmen-segmen yang terdiri dari bangun (poligon) (datar/ruang) yang semi-bebas dan seolah atau memang tidak menunjukkan adanya perulangan
- Penggunaan peta poligonal akan memberikan kesan pada pemain bahwa permainan/*game* yang sedang dimainkan seolah-olah tersusun dari dunia asimetris yang mendekati kenyataan

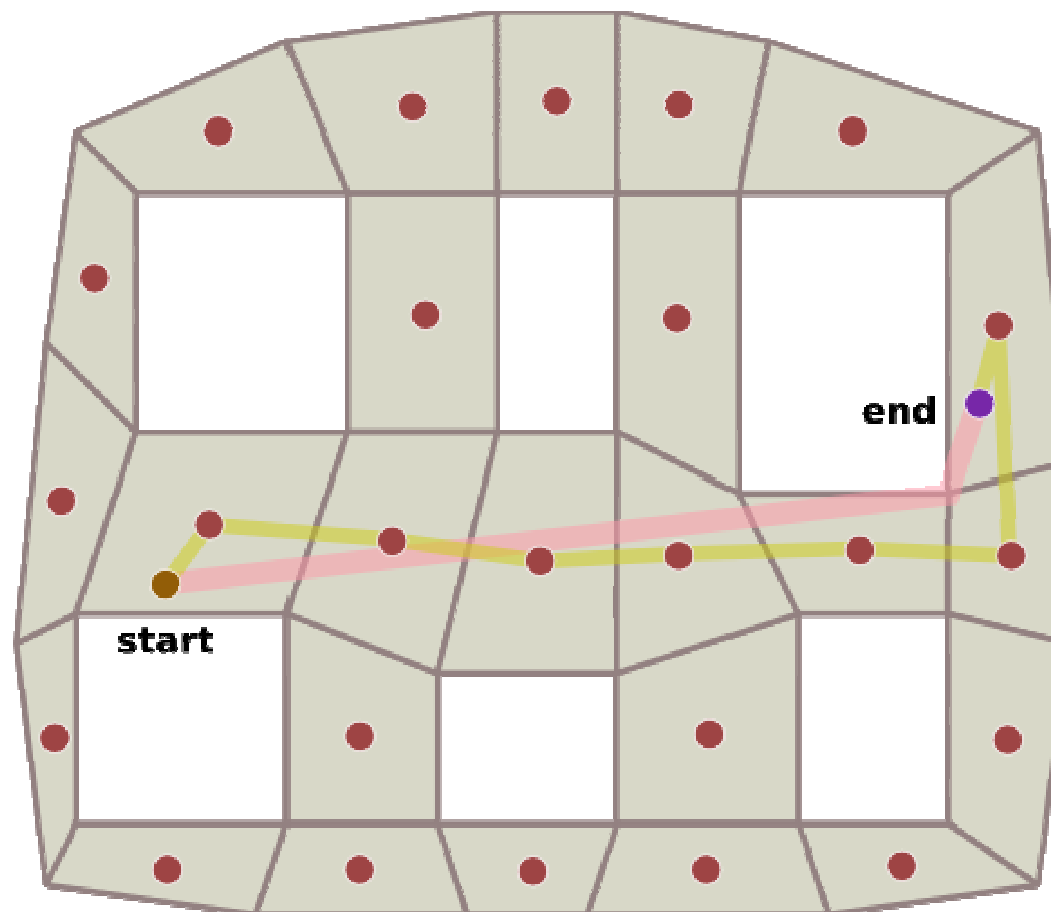
CONTOH PETA POLIGONAL

- Area peta poligonal harus didefinisikan dengan lebih kompleks dari pada grid. Ini karena koordinat titik setiap poligon dan titik pusat muka (yang pada grid tidak diperlukan) harus ditentukan sebelum bermain (bisa otomatis atau manual).



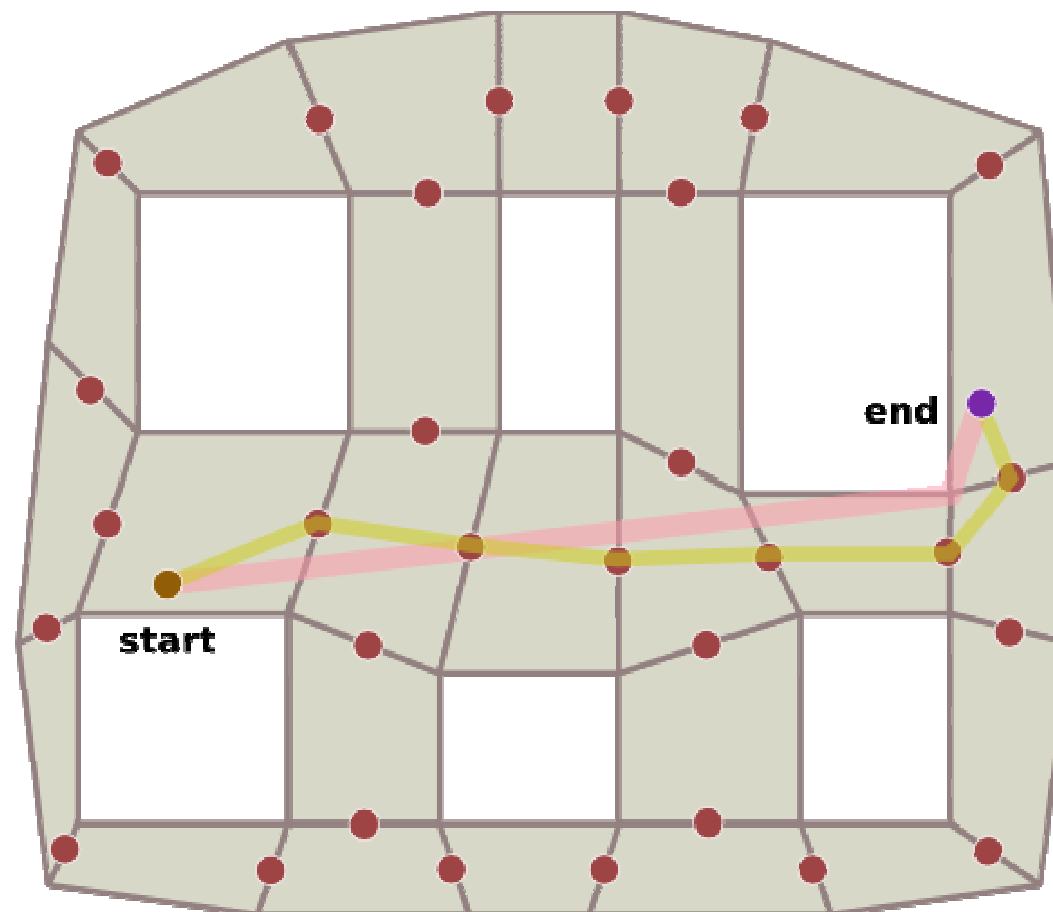
ELEMEN PETA POLIGONAL

- Titik pusat muka



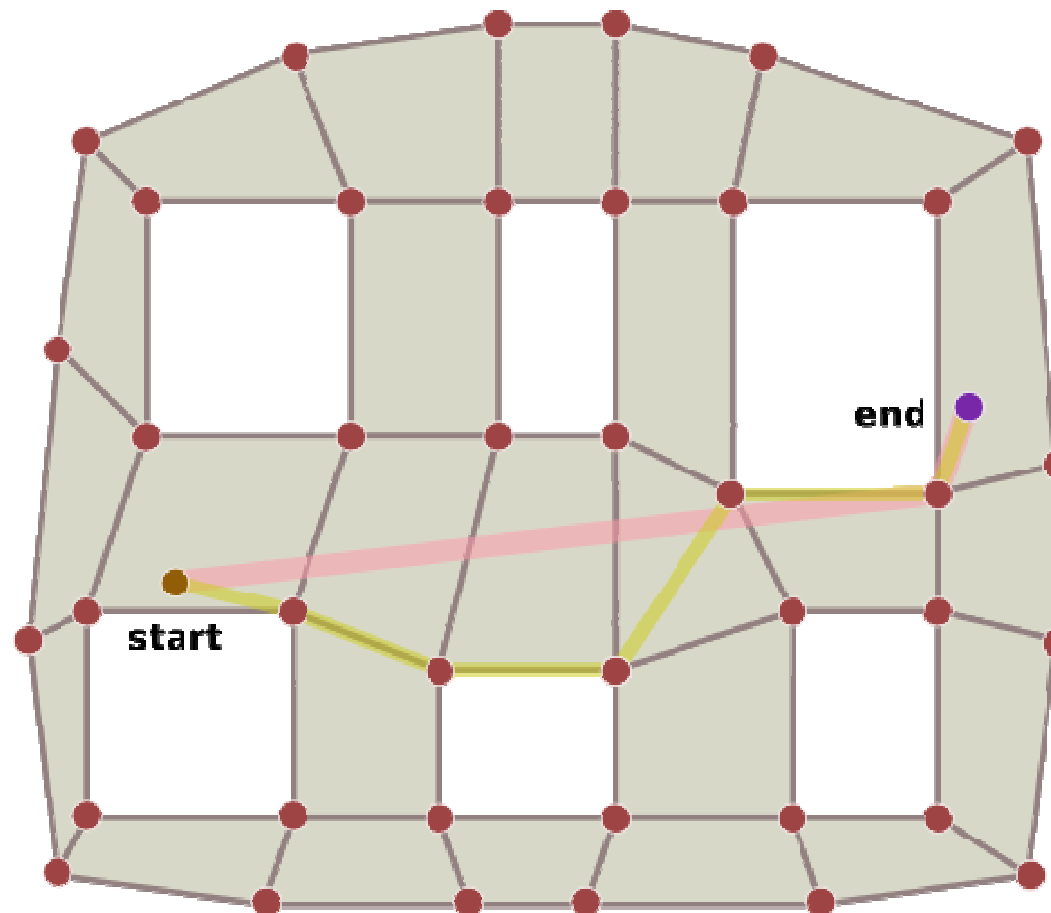
ELEMEN PETA POLIGONAL

- Tepi



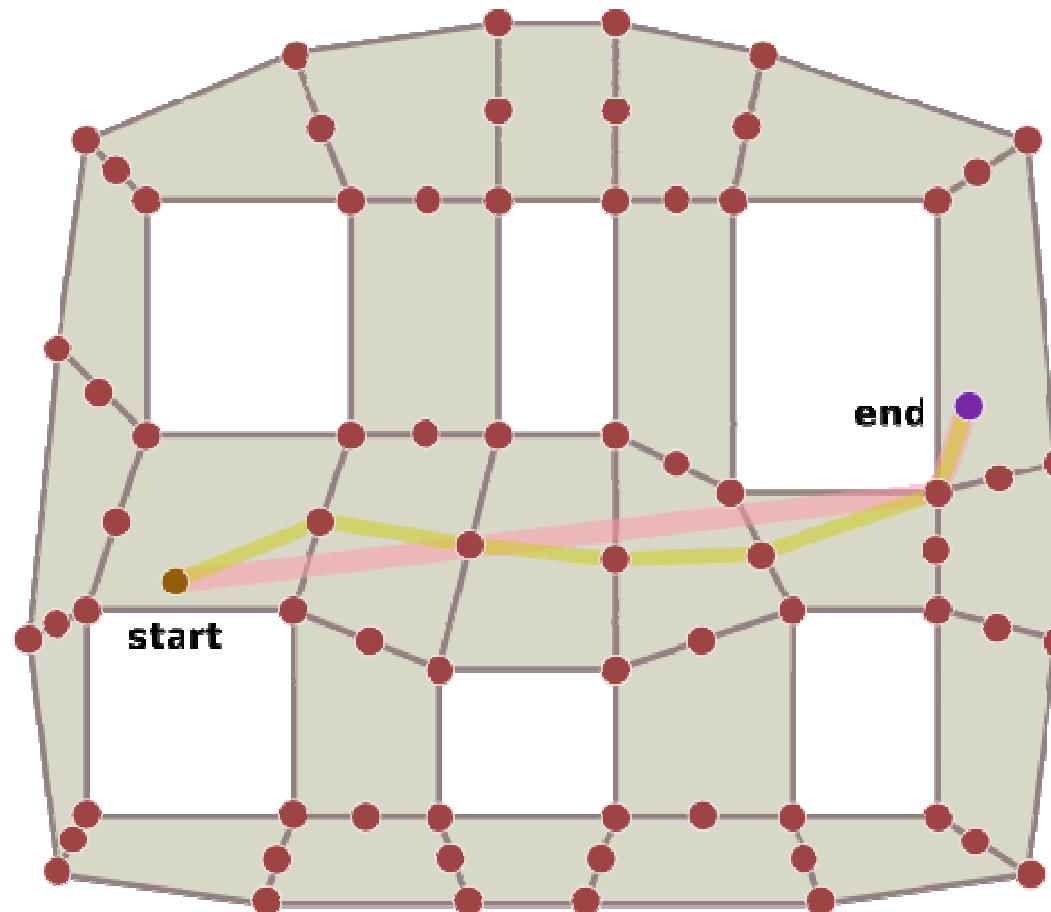
ELEMEN PETA POLIGONAL

- Titik



ELEMEN PETA POLIGONAL

- Campuran/Hybrid



POLIGONAL

- Peta poligonal lebih berorientasi pada kerumitan taktik penyelesaian jalur, sehingga butuh persiapan lebih, tapi akan menghasilkan gerakan objek yang lebih dinamis
- Penempatan objek statis pada peta poligonal didasarkan pada area poligon yang dihitung secara kasar melalui jarak sebuah edge ke titik pusat muka
- Penempatan objek dinamis biasanya tidak *zaakelijk* tetapi hanya *approximation* / bersifat pendekatan terhadap sebuah koordinat titik atau lokasi tepi

REPRESENTASI OBJEK

- Objek pada game adalah sesuatu yang ditempatkan pada papan permainan berdasarkan lokasi yang telah ditentukan oleh peta yang sudah dibuat sebelumnya (→ sebuah operasi pemetaan).
- Objek pada game dapat berupa objek statis (tidak bergerak, biasanya digunakan untuk *background* atau elemen hiasan), atau
- Dapat pula berupa objek dinamis (bergerak, biasanya digunakan untuk karakter atau elemen hiasan yang bergerak).
- Penggolongan statis dan dinamis, **terlepas dari animasi dan interaktifitasnya.**

REPRESENTASI GRAFIKA OBJEK

Secara grafik, objek dapat merupakan:

- **Objek Bitmap (SPRITE)**

Objek berupa citra bitmap (sudah dilukis sebelumnya berupa gabungan intensitas piksel). Dapat berupa lukisan *by code*, file citra statis (satu frame → BMP, JPG, GIF, PNG, TIFF) dan animasi (banyak frame → animated GIF/PNG). Gerakan diilustrasikan dengan berubahnya citra pada koordinat yang sama

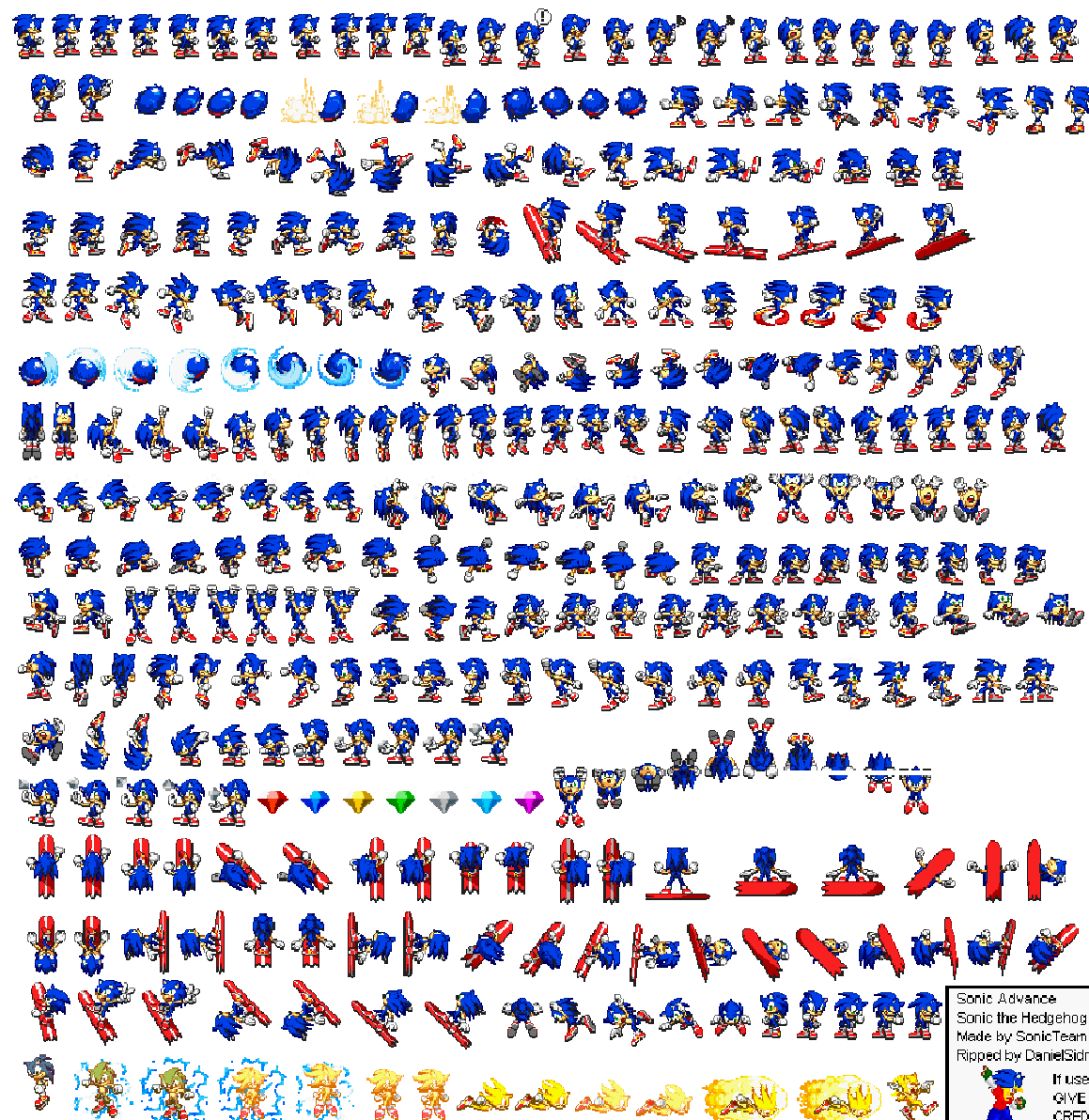
SPRITE

- In computer graphics, a **sprite** is a two-dimensional image or animation that is integrated into a larger scene. Initially including just graphical objects handled separately from the memory bitmap of a video display, this now includes various manners of graphical overlays.





SPRITE



Sonic Advance
Sonic the Hedgehog
Made by SonicTeam
Ripped by DanielSidney
If used
GIVE ME
CREDIT!
Dan.Sidney@HotMail.com

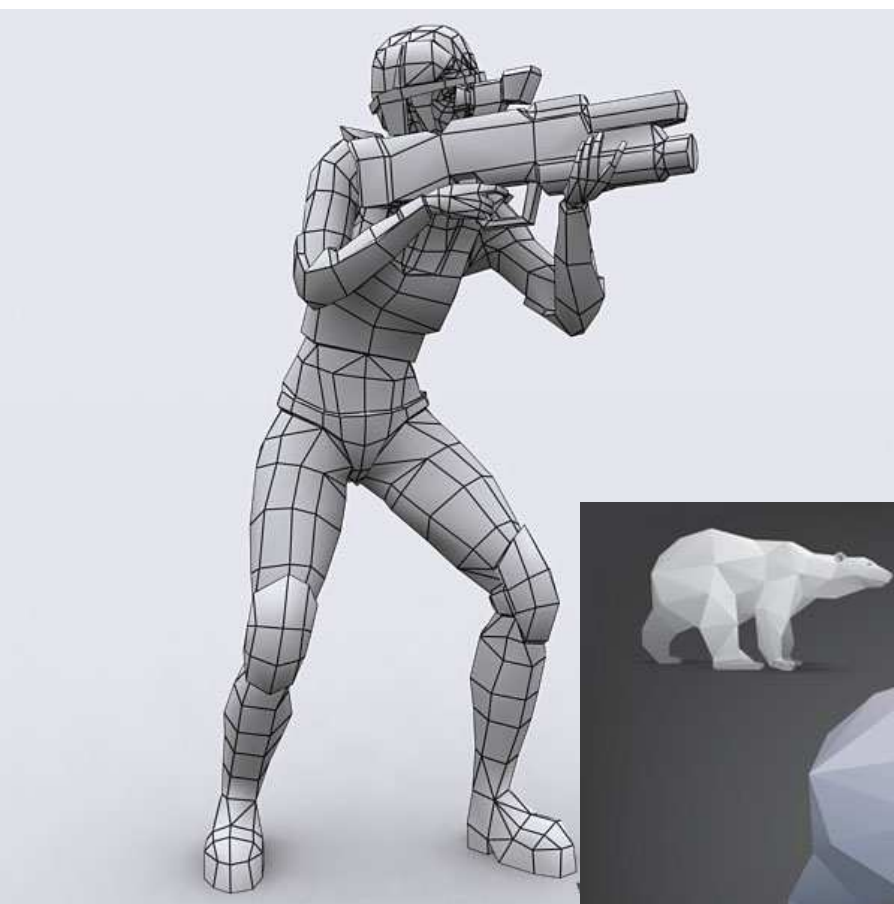
REPRESENTASI GRAFIKA OBJEK

Secara grafik, objek dapat merupakan:

- **Objek Vektor (POLIGON)**

Objek berupa data koordinat titik penyusun poligon, tekstur dan warna. Kemudian objek digambar ulang pada layar permainan setiap interval waktu tertentu. Gerakan diilustrasikan dengan berubahnya koordinat penyusun objek. Gerakan dapat lebih luwes/bebas, tetapi hasil *rendering* (perwujudan) tergantung pada kekuatan kartu grafis.

OBJEK VEKTOR (POLIGON)



BAGAIMANA PENGGAMBARAN OBJEK PADA HTML5?

- UPDATE-CLEAR-DRAW
- Setiap data pergerakan objek harus diupdate secara matematis dahulu (code), kemudian kanvas dibersihkan, lalu seluruhnya digambar ulang.
- Teknik akan dibahas minggu depan

KABAR GEMBIRA

- Sehubungan dengan diselenggarakannya ITComFest 2013 oleh prodi TI, maka bagi peserta matakuliah Grafika Game diberikan kesempatan untuk menambah nilai dengan mengikuti ITComFest:

KABAR GEMBIRA

- Submisi game ke kategori game di ITComFest akan mendapat **nilai penuh untuk semua test kecil**
- Jika menjadi finalis (tergantung sistem kompetisinya), mendapat tambahan **nilai penuh untuk proyek akhir dan presentasinya**, serta **tidak perlu melakukan presentasi di kelas**
- Jika meraih juara (tergantung sistem kompetisinya), mendapat tambahan **nilai penuh untuk TAS, proyek akhir dan presentasinya**, serta **tidak perlu melakukan presentasi di kelas**
- Detail akan diumumkan secepatnya setelah aturan ITComFest dipublikasikan



SELESAI UNTUK MATERI INI

- Materi berikutnya: TTS Dulu, kita jumpa setelah TTS di Bitmap pada HTML5 dan Menganimasikan Objek

Sumber:

1. <http://www-cs-students.stanford.edu/~amitp/game-programming/grids/#coordinates>
2. <http://theory.stanford.edu/~amitp/GameProgramming/MapRepresentations.html>
3. http://en.wikipedia.org/wiki/Tilings_of_regular_polygons (NON ESSENTIAL)
4. http://en.wikipedia.org/wiki/Sprite_%28computer_graphics%29 (NON ESSENTIAL)
5. <http://sonic.wikia.com/wiki/File:Sonicadvance1.gif>
6. <http://retrodungeon.net/?m=201203>
7. <http://www.gamedev.net/topic/451249-low-poly-opinions/>
8. <http://www.mmobomb.com/forum/general/your-opinions-on-low-poly-models-and-game-experience/>
9. <http://www.joshmast.com/wp/>