Introduction:

Street Pyghter is a fighting game. I developed it in **python** using the **pygame** library. This game is **free**, you should not have to pay for it. The **sources are open** so you should be able to look up the code if you are curious enough, if not you can download them from the **pygame** page of **Street Pyghter** (http://www.pygame.org/project-Street+pyghter-1860-3264.html).

The game resources were found on the internet, please take a look at the **credits** in game

Installation:

To verify if your pygame library is correctly installed:
in the python IDLE (or python shell) for windows user you can find it in the start menu -> programmes -> python 3.1 -> IDLE

type:
```
>>> import pygame # if you get an error here, your pygame library is not correctly installed
>>> print(pygame.version.ver)
```
it should answer: "1.9.1release-svn2575" (or a later version when it gets out)

if your pygame library is correctly installed, you can double click on Street pyghter\src\streetpyghter.py in your street pyghter folder to launch the game

for windows users:

Since version 1.2.1 there is a win32 compiled version that requires no installation, simply launch the “street pyghter.bat” file.

Otherwise get the python installer here: http://www.python.org/ftp/pyth...
and pygame here: http://pygame.org/ftp/pygame-1...

for linux users:
be careful that retrieving pygame from apt (synaptic) may get pygame for python 2 and not for python 3

Options:

The option menu lets you change the game settings. For the sound and video options, a restart is required.

Buttons:

There are only three buttons in street pyghter.

A: a quick **jab**, press it repeatedly to link them together

B: a **strong attack** to send the opponent to the ground and dealing heavy damage

C: the special button, use it when you have energy charged up, **teleports** you behind your opponent

A+B: Throw, you need to be close to the opponent, unblockable

Teleport:

You can stock up to **4 energy bars** in your energy gauge. Pressing the **C** button will let you **teleport behind** your opponent to take him by surprise.

Empty energy gauge, no stored energy bars

Half full energy gauge, 2 bars ready to use

Full energy gauge, 4 energy bars ready to use

Your gauge fills up when you are not attacking, when you hit your opponent, when your opponent blocks your attacks and when you block your opponent's attacks.

Hyper combos:

Your energy is not only used to teleport. It also allows you to launch a **hyper combo**. A hyper combo let's you land a few hits on your opponent before launching him away and teleport behind him before he can even react.

To start a **hyper combo** you need to perform a **hyper launcher**. The hyper launcher will send your opponent hurling through the screen and teleport you behind him to hit him with a jab. From there you can hit a few jabs (**A**), about 3 or 4, before using a strong attack (**B**) to launch him again.

To perform a **hyper launcher** you have to press **C** during the start up frames of your strong attack (**B**). If done correctly, you will hear a shining sound and a little golden spark will appear on your character. If you hit your opponent with a **hyper launcher** it will launch him in the air and start your **hyper combo**. During your hyper combo, your strong attacks (**B**) are **automaticaly hyper launchers** to continue your **hyper combo**. A **hyper launcher** consumes **1 energy bar**.

Hyper launchers are unblockable !

You cannot perform more than **3 hyper launchers** in a **hyper combo**. The 4th strong attack (**B**) will be a normal strong attack.

Due to damage reduce (the longer your combo is, the less damage the next move will inflict), it can be wise not to perform your **hyper combo** completely if you think you will be able to land another one later.

Making your own characters:

Street Pyghter lets you add your own characters following a few simple requirements.

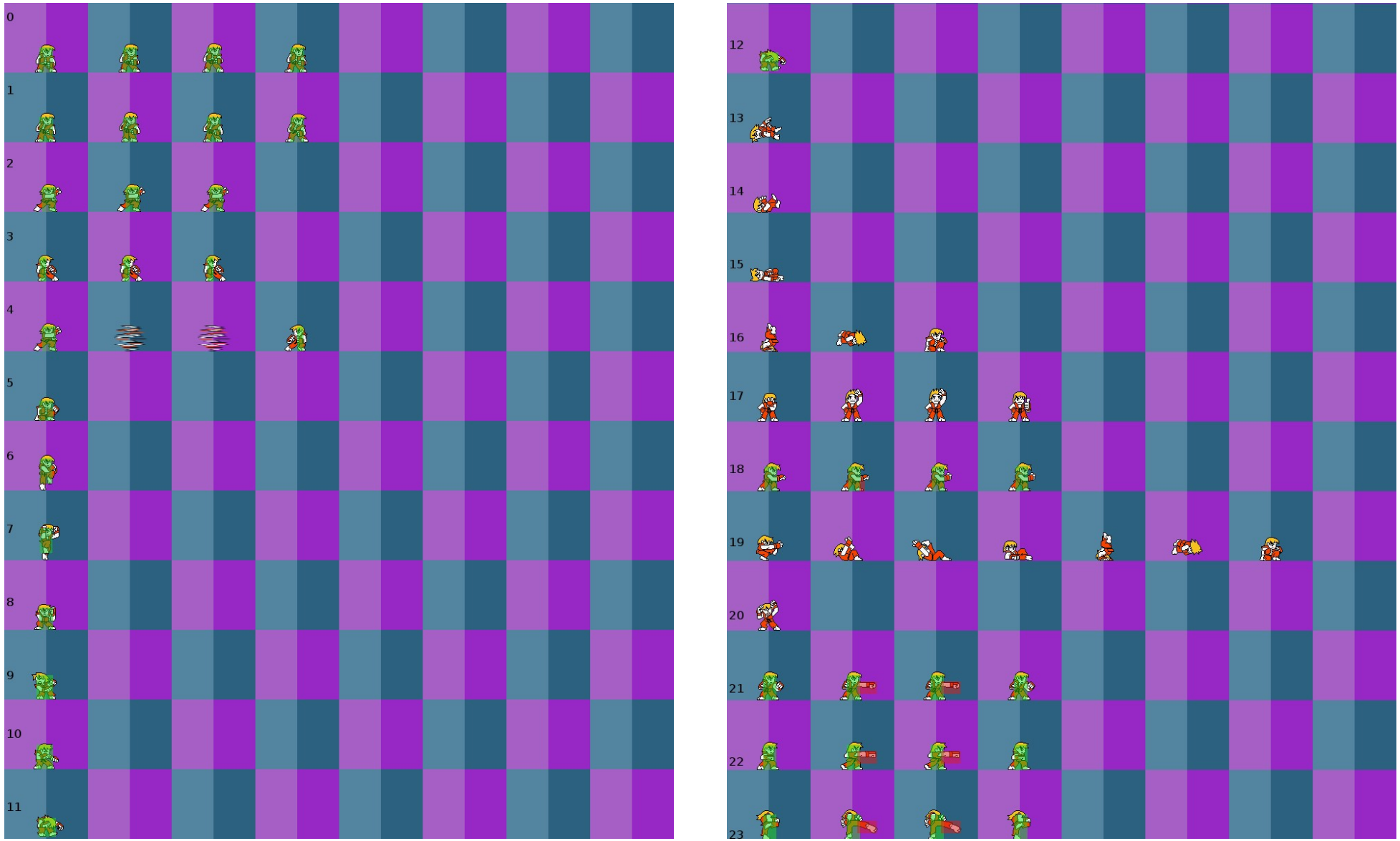
To make a character you will need to create 5 different files:

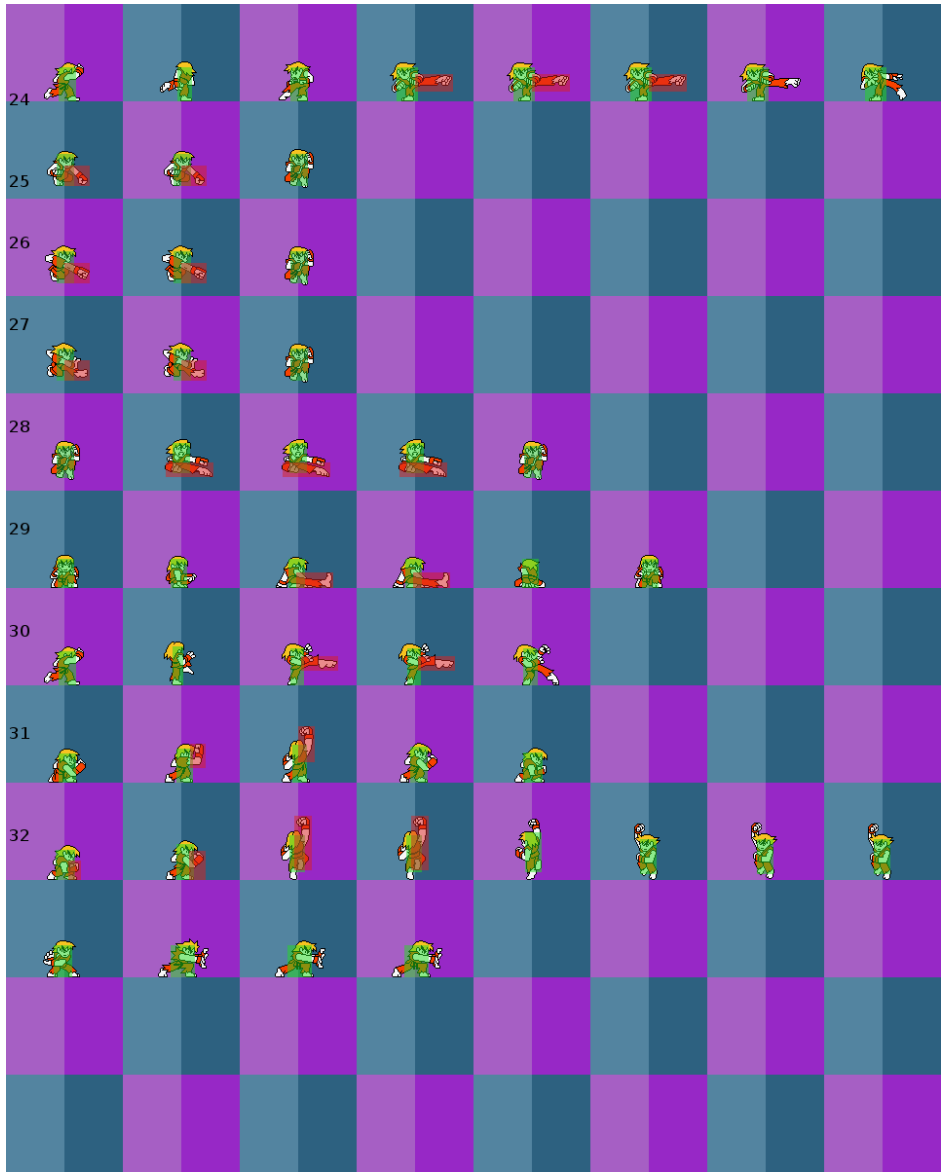
- The character's sprites for his animations [name].png. Sprites need to in 120 x 100 pixels.
- The alternate color sprites [name]Alt.png
- The hitbox sheet [name]HB.png (a rectangle for each frame to show where the character can get hit for each frame)
- the hit sheet [name]Hit.png offensive rectangle for each frame.
- And lastly, the move file, using custom code to define the character's behavior for each move.

In the folders of each character, you will also see many *.txt files. These are cache file to speed up character loading times. The game creates them if they do not yet exists. Playing a new character for the first time will take more time to load if he doesn't have his .txt files. **If you modify a character's data don't forget to erase the .txt files** or your modifications might not be taken in acknowledged.

Using a image editor program (like The Gimp, a free image editor http://www.gimp.org/) you can create a animation sheet for your character.

Example showing an animation sheet with the hitbox sheet (green) and hit sheet (red) applied over it in transparency:


0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23



Each row is an animation:

0	Idle
1	walking
2	dash
3	backdash
4	teleport (C button)
5	jumping
6	ascending jump
7	descending jump
8	blocking
9 and 10	getting hit 1
11	getting hit in air or by a strong blow (will fall down)
12	getting hit by a hyper combo
13	Falling (no more juggle hits possible)
14	bouncing off the floor
15	lying on the ground
16	getting up animation
17	win animation (stops at last frame)
18	throw (A+B)
19	throw animation if the throw connects
20	stunned by opponent's throw escape
21 to 23	light attack (A button)
24	strong attack (B button)
25 to 27	air light attack (A button during jump)
28	air strong attack (B button during jump)
29 and +	special attacks

For each characters, the animations from 0 to 28 need to be filled accordingly. All characters need those animations. For the light attacks (animation 21, 22 and 23) I suggest using the same hitbox rectangles and hit rectangles, the game **randomly takes one out of those three** when A button is pressed. Same goes for animation 25 to 27 and for animation 9 and 10.

For animations 29 and more, the effects of these attacks will be describbed in the move file.

When you distribute a character, make sure the .txt files are distributed with it, preventing your users to have to cache the caracters info themselves, wich can

be a lengthy operation.

So when you distribute a character, the folder should be the character's name, it should contain:

- [name].png
- [name].txt
- [name]Alt.png
- [name]Alt.txt
- [name]HB.png
- [name]HB.txt
- [name]Hit.png
- [name]Hit.png
- [name]Move.txt
- portrait.png (a 98x98 pixel picture for the character selection)
- mug.png (a 32x32 pixel picture for the character selection)

The move.txt file:

here is ken's move file:

Info:/Health: 1000/Speed: 2/Jump: 9/Fireballtype: 1/Hyper: 60,-60, 60,60, 60,-30/
Specials/fA:30/dA:31/dB:29/dpmfB:32/qcfA:33
fi:1--att:4,0,10,10
fi:1--att:5,0,10,10--vect:5,0/fi:2--att:6,100,6,4--vect:5,-8
None
fi:1--att:1,50,6,4
fi:1--att:1,50,6,4
fi:1--att:1,50,6,4
fi:0--move: 4,0/fi:1--move: 4,0/fi:0--att:2,90,6,4
fi:0--att:1,40,6,4--vect:-2,-2
fi:0--att:1,40,6,4--vect:-2,-2
fi:0--att:1,40,6,4--vect:-2,-2str
fi:1--att:2,80,3,3
fi:2--att:2,90,6,4--vect:-2,-5
fi:0--move: 2,0/fi:1--move: 1,0/fi:2--move: 1,0--att:1,40,4,3
fi:1--att:2,50,6,4--vect:-2,-8
fi:0--move: 2,0--att:2,90,6,5--vect:-1,-5/ fi:1--move:4,0/ fi:2--move:5,-15--att:2,50,6,5/
fi:3--move:5,-15/fi:4--att:8,0,0,0
fi:1—att:3,40,4,4--vect:6,0--point:20,-30

The first 2 lines are the characters informations: the first line determines the Max health of the character, the walking speed height of jump, type of fireball (sprite) and the vectors of the hyper combos (in which directions the enemy is sent). The second lines determines the motions to input for special attacks (the move list).

The hyper combo vectors are listed as follow “Hyper: x1, y1, x2, y2, x3, y3”. Where the vectors are (x1, y1) for the first launcher, (x2, y3) for the follow up launcher, (x3, y3) for the last launcher of the combo. After the last launcher any strong attack will just end the hyper combo.

The Second line determines the motion for each special attack, here is a few exemples:

fA	forward + A attack
bB	back + B attack
dA	down + A attack
qcfA	Quarter circle forward (down, forward motion) + A button
qcbB	Quarter circle back (down, back motion) + B button
aqcfA	Quarter circle forward in air + A button
aqcbA	Quarter circle back in air + A button
chduA	Charge down up + A button
chbfB	Charge back forward + B button
hcfB	Half circle forward (back, down, forward) + B button
hcbB	Half circle back (forward, down, back) + B button
dpmfB	Dragon punch motion forward (forward, down, forward) + B button
dpmbA	Dragon punch motion back (back, down, back) + A button
360B	360 motion (exemple: back, down, forward, up) but works in both ways and from any starting direction + B buttton

For each motion you define the animation it corresponds to.

The next lines are the effects for each animation. They start at the 18th animation.

- “fi:X” determines at which frame of the animation theese informations are for.
- “/” is the separator between frame informations
- “--” is the separator between informations of the same frame
- “att: X, D, A, B” determines it is an attack: X is the type of the attack (see the table below for the different types and their differences) D is the damage the attack does, A is the hit stun (time during which the enemy will be in animation 9 or 10 if it's a light hit), B is the block stun (time during which the enemy will be frozen in animation 8 if the attack has been blocked)

HIT_TYPE_NONE = 0	empty the attacking buffer (rarely used)
HIT_TYPE_LIGHT = 1	light attack, enemy stays up
HIT_TYPE_HARD = 2	hard hit, enemy will fall to the ground, a vector can be associated to determine in which direction the enemy is ejected
HIT_TYPE_FIREBALL = 3	creates a fireball, a vect information is required for the firaball moving vector
HIT_TYPE_THROW = 4	a throw, cannot be blocked, jumps to the next animation if it connects
HIT_TYPE_ALWAYS = 5	a sure hit, used to move the enemy during a throw without causing damage
HIT_TYPE_EJECT = 6	used to end a throw a vector is required to determine in which direction the enemy is thrown
HIT_TYPE_HYPER = 7	used to start a hyper combo (rarely used for a special move)
START_FALL = 8	reinit the movevect of a fall

- “move: X,Y” allows you to add a movement vector to your character during the attack animation.
- “Vect: X,Y” used to give a vector parameter to the attack depending on it's type.

It is important to follow the order of those lines:

line number		animation defined
1	character informations	none
2	move list of the character	none
3	throw attack	18
4	throw when it connects	19
5	# None (animation when enemy breaks from throw)	20
6	light attack (A button)	21
7	light attack (A button) should be the same as above	22
8	light attack (A button) should be the same as above	23
9	strong attack (B button)	24
10	air light attack (A button during jump)	25
11	air light attack (A button during jump) should be the same as above	26
12	air light attack (A button during jump) should be the same as above	27
13	air strong attack (B button during jump)	28
14	informations for 29th animation	29
15	informations for 30th animation	30
16	informations for 31th animation	31
17	informations for 32th animation	32
…	…	…

Last advices:

Remember that following a throw attack is always the throw animation if it connects.

When creating the rectangles of the hitboxes or hits, make sure that they do not end at a border of the whole image. If they do, it is always an option to add a line or column to the animation. Empty frames will not be part of the animation and will keep the rectangle away from the border.

Move lists:

Ken:

▼, A	Uppercut
▼, B	Low sweep
►, A	Reaching kick
▼, ►, A	Hadoken
►, ▼, ►, B	Shoryuken

Rick:

▼, A	Uppercut
◄, B	Heavy punch
▼, ◄, A	Thousand knuckles
▼, ►, B	Comet punch
►, ▼, ►, B	Cross punch

Leona:

►, A	Double bite
(in air) ▼, ►, A	Sonic boom
(in air) ▼, ◄, B	Dive cut
(in air) ▼, ►, B	Moon kick
▼(charge), ▲, B	Moon cut
▼,►,B	Wind cut

Zangief:

▼, ►, A	Dashing headbutt
►, ▼, ◄, B	Spinning lariat
360, A	Pile driver
360, B	German suplex