

Breakers Gambit

Overview

Basically I've always wanted to make a 2d fighting game. Been playing fighting games with my bro longer then I can remember, so you could say we've done our fair share of research. Knowing the type of game play we love, I plan to make a 2d fighter myself in C++. Obviously I cant do everything myself, so I'll do the most part of it that I can, with assistance of others.



Architecture

After learning a lot of C++ at uni and not touching anything graphical with it, I decided to look up how. Found and experimented with the SDL library. Turns out that should be enough, with clever coding, to make my fighter. So mainly working with the SDL lib to make this one.

Components

SpriteBase

This is the class that stores all the sprite info. Like animation, transparent colour and delay time. Most of the examples I've screwed with have used such a class, I just need to alter it to take more then just this.

First alteration would be it to store versions of the animation. One which is normal, the second is horizontally reversed. This saves flipping surfaces on the spot. I already have a surface flipper, I just need to cater for it within the SpriteBase class.

Next would be letting it set the origin point of the animation aswell. This is the point which helps place it in a sense that makes sense. Mainly will be placed at the feet or centre, to figure out how the object interacts with the ground. These will be ofX and ofY. Which stands for orignal feet x,y.

Lastly, the bounding boxes used for collisions. This will have to be different per frame, so this is the best place to store it. This will be in the form cx1, cx2, cy1, cy2. This is in relation to the sprites X and Y, and not the ofX and ofY.

Fighter

This is derived from the sprite class, but with a lot more added. More on this part later

Joshes part?

This part figures out if any combos have been done from the players command list. If so, return which move was activated. A lot more detail on this later. But basically after a move is returned, it tells a fighter a class what move to do. The current move the fighter is doing is cancelled and the new move overtakes(unless a special move is already in progress).

Attack

This is all the info on 1 attack, ie how much damage, damage collision box, character movement, projectile, special effects etc.

Command List

This stores all of the special move button combinations and 'Attack' classes associated with each attack. Essentially a list, with more priority on the attacks at the front of the list. The higher the priority, usually the bigger the attack.

Special effects/Projectiles

These classes are drawn last, as most will take place on top. These generally die at the end of their animation, thus need checking regularly to see if this has happened. Projectiles or damaging objects will more or less be derived from the special effects class, but check for collisions against said opponents.

Diary

19th Nov – Started this document today. Hard to motivate myself to get started. Already have Josh doing a component for me. Will start soon, but I have a lot of things I'm meant to be working on.

20th Nov – Got SpriteBase class done pretty much. It can flip images and store collision boxes, as well as origin points. Also started with my fighter class, which is derived from Sprite. So far he can face the direction he moves in.

21st Nov – Fixed up movement and added a basic jump feature.

23rd Nov – added double jump, woot. Some planning now –

Mission: Multiple animations

Easy! Sort of... Make an animation class set which stores sprite bases. Then make the fighter class point to such an animation set. Whenever the animation changes, fighter class changes its pointer to a different animation and also changes the bgreplacer variable thingy. Too easy. ALSO, should learn about resource files.



26th Nov – Looked up resource files. Looks easy enough. Also, thinking I will have to store animations in their own animation set class. A standard one which just stores animations, then classes derived from that one which hold/load specific sets of animations using resource files. Also, with that I would need a class which holds a set of collision boxes so that multiple frames can have different hit boxes.

27th Nov – Basic set should compose of

- Standing
- Walking
- Running
- Walking backwards

- Short back jump
- Jumping in air
- Ducking



7th February – I never spell this month right. Its been awhile, but I've been looking at resourcing. For some strange reason the way I'm resourcing, the resource file screws up with too many files AND for some reason, the image either doesn't load or screws up if the BMP isn't 8 bit, and have a standard 256 palette. This isn't to big of a problem however, seeing as fighter sprites tend to only try max out on 16 colours and theres a decent amount of standard 256 palettes to get what I want. Just getting Oliver to change/manipulate my resourcing program. Why him? I cant be bothered.

Success! So now I have a program which will make resource files out of a whole directory worth of files, and I made a pretty simple function for loading up the resource. This will help things along greatly. I'm quite happy right now. This is a success. I'm making a note

Ideas department. A sprite needs info per frame of its animation. I need to factor in 2 things more I think for sprites. Frame by Frame Damage Collision Boxes and also frame by frame Sound effects. This way I can have a combo in just 1 animation, and since the sound effects are called frame by frame, this means the combo can be interrupted fairly in the middle.

Also, I will need 2 new classes. The player class, which accepts input and interacts with their fighter, and also a AI class, which controls the enemy fighter. The AI class is outside of the fighter class, so that it can view everything happening on the field. More or less, the AI class moves its fighter like a puppet. This is a lot better, as the AI class can send info to its fighter class in the fashion that the player would. AI class could easily be manipulated to make easy, medium, hard etc.

Fighter classes will need a couple more variables and things. They need a variable saying if they are busy or not, so they cant make attacks whilst doing an attack. Although, a certain amount of queuing wouldn't hurt. Also need a variable which holds the status of the player. IE falling, lying on ground, normal etc.

12th February – Something to note, and really should be listed somewhere. The 'info' file that sits with resources **NEW** format will be of:

Files: [number of frames] ofx ofy

```

colx1 coly1 colx2 coly2
#comments
image.dat      [pause time]  r b g[alpgha colour]  d.X d.Y d.X2 d.Y2 [damage]
image.dat      [pause time]  r b g[alpgha colour]  d.X d.Y d.X2 d.Y2 [damage]
etc..

```

And... Yatta! I have an animation loading from a set of resourced files, following the rules given from the said info file. Next step is to setup animation sets.

February 4th – Alright, since I barely do any work. Today all I'm doing is planning. In the Data folder, there will be a 'Character' file, which has the number and all the names of the characters. Using the names, it will be able to find the animation set files. One will be in the character file, the next one in character/attack and finally one in character/damage

An animation set is passed a character name, then stores all the info from said directories.

February 22nd – sndFX has been added to spritebase, but I doesn't read in the sounds yet.

September 7th – Its been a hell of a long time, so I need to put down quite abit. In my absence I've found out a few things and have learnt them to a degree that I can take advantage of them. Turns out SDL stuff is rather slow and uses a lot of CPU power, so I have looked into SDL+OGL and have finally gotten it to the point where I can show a sprite in opengl. This is very good, it'll run a lot faster now. Theres a few more things I need to test however. I need to see if holding 100 or so surfaces is slower then holding 100 or so textures. More on this later.

Is now later. Did a simple storage test. It would not be a good idea to have all the surfaces loaded in all of the time. But I guess I didn't test deleting the surfaces and keeping the textures... one minute. Nevermind. Waiting for 1000 textures to be loaded in takes awhile, its not fun. Also uses up quite abit of mem usage. Load up where necessary then.



6 January 09 – First diary entry of this year. Progress is always slow, but heres an update. SDLOGL dropped. Not feasible effort wise and somewhat pointless. Reading in heaps of info per frame of character, using resourced stuff of course. Built an animation storage class for basic animations and also a moveset. Heaps similar but also holds extra info on how to perform these moves. Now I have to make my fighter class use all of this. Prob need a sound and particle class aswell soon.

20 January – Got a debug thing working, so I can see offset and collision boxes. Should be easy enough to show damage box later. Have my little guy change from walking to jumping now, using the animation class. Fixed a

bug with leaving animation change images behind. Just need a 'blank' animation which is larger than all animations. Miiight do some more tonight.

31 January – Got punching working, huya! Duck got working the other day aswell. Getting there. I'll get the damage box showing in debug next, so I can check that too.

3 February – Ok, got damage box debug view working. So its in the right spot. Noticed the drawing of the character wasn't in the right place, not at the right offset. So I've corrected that also. Now he moves rather nicely around. Go team!

10 February – Added the entire inputChecker class. Haven't tested, so there's a good chance it wont work at all. Perhaps a lot needs rewriting. Testing can only tell. Well, its main function is to read the input of the player and see if he's done any specific moves or attacks. For example Ryu Hadouken from Street Fighter, down, forward, punch. I need a class to read and decide if that falls from the command list.

12 February – Hella yes!!! Well that took ages. 2 days ago all the linked lists I did were pretty shitty, so it took ages getting everything working. But now my input checker class kicks ass. My bro's little puffindairy character can punch and flaming head butt. Need to add more input and something to determine if your facing left, to still input a 'R' for right. That way it can still read the same commands.



28 February – Today added more input into the checker. Ie it now adds left right, up etc. Not the other attack keys yet though. Also made inputChecker take a fighter pointer. But its gonna need a target pointer when I make it. Also made it so if certain frames make you move, it now does that. For example, the flaming punch makes Puffindairy jump and move across the screen in one move. Hard to explain.

Need to get started on conditions class and target class. Shouldn't be too hard?

5 March – I don't remember when I did the condition and target class, but I did them last week abit. Today I made the conditions class return if a player is on the ground or not. Then tested it. Was to unmotivated by motivation to do more. Need to add it to the input checker class now.

6 March – Tonight I added the Conditon class to my input checker. At the moment, the only condition it checks, is whether or not it the fighter is on the ground. So now he can only punch when he's on the ground. I can add conditions whenever I want now. I think I'll make my next mission an aerial kick. Make sense ya?

Ok, done aerial kick. But can fly with it. Not good haha.

10 March – Well I made you don't fly with kicks now. But it feels abit ugly... Maybe, you use up some sort of jump quota and then if you do the attack again, you just drop. Not sure, needs more thought.

17 March – Blast, seems my timer isn't accurate. I need to adapt to the old one I suppose. I'm not sure why, but it works better. Its gonna be a bitch converting this,

but it will move smoother. $2.5*dt$ is the min speed I can see so far however. Maybe I can just check if that amount of time is passed as a minimum somehow? I don't know, I'll figure it out.

19 March – Well the old timer system didn't work with my current system. Somehow this game runs the code faster. So the timer in the other example doesn't work. Which is weird, its suppose to cater for all systems. Oh well, I wrote up a system using an accumulator. The prob before was that my while loop, gets ran every 2 milliseconds. So even moving him a pixel is an insane amount when you think about it. Now I have an accumulator, it adds up to 10 ms and then the codes allowed to adjust position values. All good, greater control, me happy.

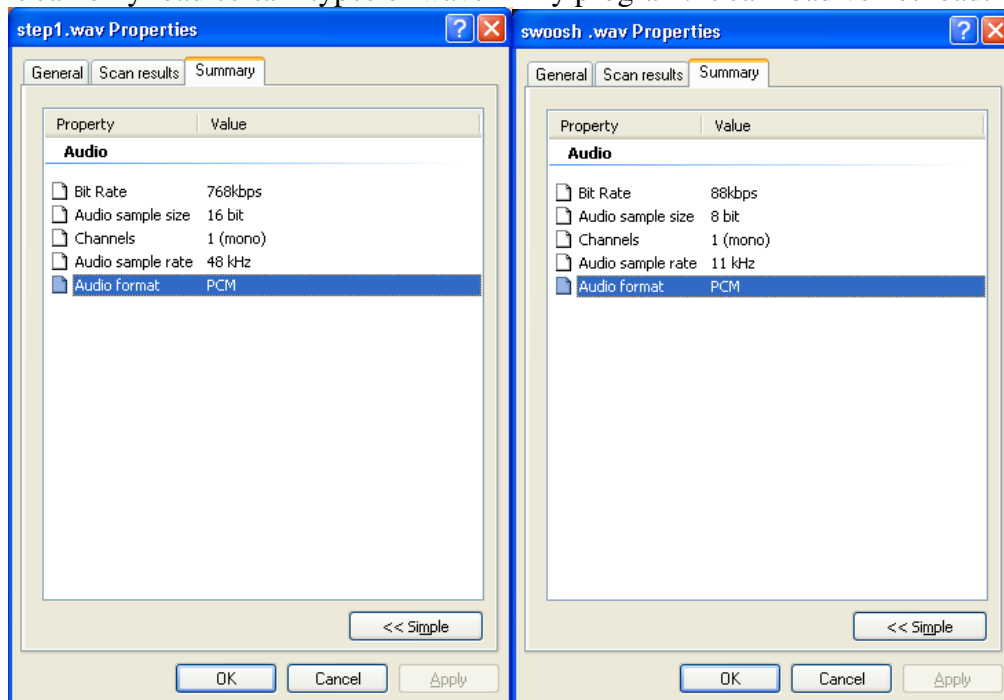


26 March – Added Ken from Pocket Fighter to replace Puffindairy. All the sprites are already done, so testing is easier. Had a problem with doing moves facing left, as it used the x,y offset of the right direction. I fixed this and left over images from turning around. Starting to look nicer.

Havent updated the debugging boxes for looking left yet.

27 April – Its been over a month already, god time flies. Well I've started working on the fabled sound class. Its mostly complete, just needs to be able to store more sounds and use a 'find' function to be able to find the sounds that are in it. Gotta update the main code too to support when and when not to play a sound, otherwise it'll play a million times at once. Its half in, just more to go.

1st May – Almost completed my sound class. Mostly happy with it, not happy with the internet not having as many free sounds as it used to. GRRR. Anyway, turns out that it can only load certain types of ways in my program. It can load vs not load:



Odd yeah? Gonna finish sound class now. Next task is particle class, which includes fireball attacks and stuff.

Ok, sound class done. You can add multiple sound sets together now. I was going to make it so you couldn't add the same sound twice but meh.

Now that's fixed aswell, hell yeah! So now it cuts out reused sounds when your adding to a list.

2 May – Started particle header. Too sick to continue, no ground really made. Particles will also be used as damaging objects, such as fireball attacks aswell. Will elaborate later... tired.

3 May – Kept working on the particle class. It kind of works, however when drawn it seems to not replace the background behind it. I'm not sure why exactly. Too bothered to bother now. So really, from here I need a particle set class and a particle holder class. One holds the definitions, the other collects particles created and draws and updates them correctly. More testing on the particle class itself, as it needs to be able to die and stuff.

In testing all this, I have come across a series of other bugs. I'll list them below so far:

- Screen boundry needs to collide with mx not body size
- Make a standing animation and make it work with everything
- Debug doesnt work for Left
- Boundary edge checking doesnt work exactly in fighter anymore
- If x,y offsets change mid animation, it screws the background updating
- Currently particle class doesn't replace background properly

That's all for tonight I'm afraid.

15 May – I come back a week or 2 later, and its not working. How lame is that. I think I'll remake the particle class from scratch. Its too buggy.

22 May – Ok, well I didn't do any work today, but I did on Tuesday. Basically remade the particle class. Having it as a subclass to the sprite class was pointless and buggy before. None of the image loading part changes, so I went with another approach we learnt from my comp sci degree. I don't remember its name, but I basically made a particle class hold a sprite class and have its own correct information. Voila!

So now I need a particle List, so that newly created particles can be drawn on the screen and killed off accordingly. Fun? Perhaps. Only if I get it working.

24 May – Alright, so I've added more to particle class and made particleList class. Now we can add particles to the list and they will be drawn and basic movement updated. That's pretty decent so far. Now to add particle removal and the rest of the particle features. Also need to make it so you can create particles on attacks/moves.

So now the particles can be killed if they go off screen. Congrats me! Next step is making them be created when needed, which is more effort...So I have particles

stored as strings in the animation frames themselves. They are in the format particle name, x and y. So I need to break it out of the string and make it. Yaaaaaaaay. Not for tonight.

1 June – Well I did some work on this earlier last week, but was not at home to update this part. Well I worked on getting the particles to be able to be alpha'd. So now if they need to, particles can be slightly transparent. This is applied across the whole service. I don't really care about partial transparencies on certain colours or alpha maps. Screw that.

Also I made it so when particles are created their position is transformed to the correct location. This is very good, now particles will appear in the right location. Next I have to make it so players can shoot or make these particles.

Today however, I spent ages tracking down a SDL_net example, which I had no luck with. So I worked on making the default examples work. So now I have a TCP/IP connection working, next is the UDP one and then multiserver so multiple people can join. How exciting! This is important later for when players want to fight across the net or if I ever bother making a mmorpg.

I originally had a problem along the lines of 'sdl.dll sdl_strlcat can not locate', the solution was to make sure the right SDL dll. Was in the folder, it had nothing to do with the sdl_net.dll.

More today. Trying to make it so particles can make the right data based on a string. Thanks to Josh helping me figure something out after me being a noob, progress can continue. Must remember variables made in functions die after the function ends and pointers cant just point at them, so now I need to figure out how to store enough info on particles on the fly... Give me a few minutes.

A lot of fucking around and not a lot more gained from this experience... So now we can create particles on the fly but it's not right. I can only create 1, which gets stored in my head pointer in my list, any others are just temporary memory and die after the loop or function ends. So annoying, I need to be able to store new particles as they are created without making arrays or anything lame like that. Maybe I'll get time later this week... We shall see!

5 June – This particle list is giving me no end of trouble. I'm far beyond annoyed, something so simple shouldn't be so hard. I managed to get it to work half on my laptop, a vista machine but porting it back to XP it just crashes. Memory must be dealt with differently in vista... I may convert this crap to use list.h, it may work better.

Interesting how xp and vista run my game differently. Not half bad testing on both. After I get prototype 1 done, I'll try other OS's.

8 June – Fuck particles piss me off!!! Gutting them out of my project and starting a fresh sounds like an awesome idea...

5 July – No progress so far on the code work unfortunately. However, the sites been updated with a forum so people can fight the good fight and help me. I've screwed



around with the tracing pad and did the very first frame of Burn's starting stance. Need to adjust his hair smaller in the other frames, then I'll pixel the rest up. Then release it again so people can help me with it all!

8 July – Salvageable, its all salvageable. Look back through the particle class stuff and what I've been thinking about, it seems I can save it yet. Since storing images on the fly and opening a lot of txt files and image files causes it to crash at random points. Well there was never a problem if the particles data was preloaded. So instead of just making particles I should make a particle data list, which stores all necessary info and particles only reference this info, instead of creating it. It'll take a lot more time to setup, but totally worth it.

I would much rather have done all the particle stuff on the fly, as now I have to store all the particles until their use is no longer necessary. This is good though! I can do it. Also, I have finished the first animation of Burn doing his start pose. Soon to be made a gif somehow. How exciting!

17th July – Over a month from me being extremely pissed off, I've come to the point where I am extremely happy! What's happening you say? Lots! Ok, firstly, I just got the particles to work. They get created at some arbitrary spot on the screen, meh, but the main point is they get created and stored at runtime and all work and display and happy. I changed my entire approach. So instead of loading everything from files whilst the battle rages I have all the images preloaded into a list, and then any particles created reference these images and info and then those particles stored in another list. Hell yeah bitches and that works.

More news? Yes there's a lot! Slim pickings getting random net goes to join, but atleast I have Adam Parkes. I met him at uni, and he works fucking fast. He's currently working on a basic tcp server/client example. He's working really well on that, it's only been a week and he's almost finished the example. After he has done that I will let him help on the main game itself. He is very efficient! Also, as of tonight I have grabbed Trav to design the title of the game. This will be seen a lot, I wouldn't trust just anyone to do this. I expect great things.

And finally, I have the first pose animation done for Burn. Havent had a lot of time to test the pose unfortunately. I have a few images and screenshots now I want to upload to my site. Things are getting onwards forwards.

26th July – Another 10 days or so have gone by. Well in that time a guy called Rodrigo said he would join in on the pixelling part of things. So far he hasn't done any work, but he's got me thinking about the requirements for the sprites themselves. Which lead to creating the Sprite Requirements Document. Which will be downloadable of course from the website.

A new Burn sprite was created today based off of the new requirements. Here he is.



Cute right!

31st July – First of the month tomorrow. I cant believe its almost my birthday and I didn't even know. That's how busy I am. Well, Mostly did art and job interviews this week, but I've made little Ken shoot a hadouken properly now. Then fixed a few bugs. Some small group of pixel artists bugged me for payment to join my project, I don't think I will accept them. There isn't a lot of pixel work to do at the moment anyway. Well I'm tired, I'm going to release a demo soon with more bug fixes. Night

16th August – When particles die, they have the ability to create more particles from themselves. This will be useful for explosions etc. Particles can die by falling off the screen edges or by their timer killing them. More needs to be added, but that's as much as I can do with particles at the moment. Onto the next class, which will be the control set of classes. Control classes help decide how a fighter moves etc whether through player interaction or AI. Need to write this up. Also will need a collision class soon. Exciting!

25th August – Coding hasn't progressed much. Made a basic game pad class. That will need fixing up later once I figure that out, but other then that nothing. I've done work on the Burn sprites though. I'll just post the main stance up. Soon to work on jump, duck, block and a few attacks. Then a basic taking damage image and then get that working. Once that's up, I'll release a new demo. With the planned code that is.



26th August – All right, done the animations for the basic movements. Standing, jumping, ducking and walking. Need running and a whole bunch of blocking, attacks and damage images. Perhaps I'll build a basic chart I can print out for all characters I build so I can keep track of everything I need. Demo will be out soon. Need to manually make Burn up as a playable character.

2nd September – Well worked on control and playerControl classes. The player control is half up, it's at least enough to replace the code in main with its update function. Need to make gamepad class usable. That would make me pretty happy. It looks pretty easy to make a simple program to setup the keys for input aswell. Oh yes, I have to mention this its important!



NOTE: Poll events only works if your doing it with 1 poll. Multiple polls does not work. So if I have multiple playerControls I need to make it so they have a shared events polling system for input. It's a good thing to know.

4th September – So tired. Tried making it so my gamepad class would have the default keys set, but then the game freezes weird. So if I do it again, it does the same thing. Not really sure how to get around this. Kinda tired... Sleepy. Agh! I was wrong. Turns out the SDLKey things in that class weren't making the game freeze weird, it was actually (for some reason) the Uint8 types. Commenting those out for now will do, since I was using those for actual gamepads but I'm currently focusing on the keyboard input for now.

char *SDL_GetKeyName(SDLKey key); Looks useful for later on for displaying names of entered keys. However, at the moment I have no idea how to set and change keys in code. Also, I can pass around the events type so I don't have to have a function sharing 2 players on one machines input. Should have figured that out myself. Ugh, seems my key problem was my own stupidity. In my input checking part for the B button, I accidently had it checking the gamepads A button. So since my current fighter doesn't have any attacks with the A button, I would never have pressed A. So its all fixed now.

7th September – Perhaps another day wasted finding things out about things that don't get me further on my game. I had a class, but when I wanted to add more variables to it the game would freeze weird. Didn't help being a struct either or being called something else. After many hours of research and what not, it turns out I somehow have too many variables in Main. This is ok though, as I'm trying to cut main down anyway, so this isn't a problem. I can just chuck gamepad class in as a global until main is smallerized. Also learnt I can check keys using the keys name instead. This way I can save and load keys as strings. Rather handy. Also learnt a lot about joysticks and they seem easy enough. I have decent documentation on that now. Also I have the input checking function get passed the SDL_event. This way I can actually have multiple players quite easily! More on gamepads and such tomorrow I think.

8th September – Ok, so it looks like the SDLKey type is an enum, which is awesome so I can just save in and out the keys as ints. To easy. Adding a second player has issues haha. This needs a lot of work. If I could find my brothers gamepad I'd set that up. Randomly looked at music. So we need to:

```
Mix_Music *music = NULL; //initialising
music = Mix_LoadMUS("music.ogg"); //loading
Mix_PlayMusic(music, 0); //second arg: loop amount. -1 for infinite
```

```
Mix_HaltMusic(); //stop music  
Mix_FreeMusic(music); //free  
music = NULL; //set to null
```

Adding Burn as a playable character now. He's quite large haha, I thought I had to drop the screen size to 640 x 480, but that would be too small to battle in. Going to keep the 800 x 600. Awesome, so now Burn can move and jump around and generally have fun. Going to find a new background image and release a little demo. One of dad's beach photos will do.

8th September – Added joystick capability to it. I can't believe how easy it was to add this to my game. Definitely gonna back everything up now. So you can walk around and do the various attack combos with the joystick/gamepad. It all works, thing is I haven't added the fact that you can setup the keys in game. This will have to come later. The next adventure is to have an enemy up. Firstly to check collisions and then secondly, to have some AI running. Need to add power and hp to the fighter class. Might do that now.

13th September – During the week I made an AI class. All it does is drop the fighter with gravity. Also fixed the debug drawing bug, now it draws the collision boxes correctly. Made it also that you can target the enemy and face them constantly. Now onto collisions. Using Joshes SquareCollider class I have basic collision between the 2 fighters. Now they can't stand on top of one another. Suppose the next one is to check damage or something?

14th September – Alright. So I added damage collision checking between the two fighters. Originally I had the AI just constantly throwing punches for you to get hurt by, then copied that code over to the AI. I made a hitRef class, so that I don't have to keep making a reference function to see how to take damage from certain events. So now when this one gets updated, everyone will be updated. The AI now looks at you whilst you pummel the hell out of him.

Next I need to add damage collision with particles. Not heaps of work, just tired to do it now. They need to get the damage collision box. I just need a system to see if my main collision box collides with any of the particles.

26th September – Ok, now as of today you can take damage from the enemies Hadouken attack. This needed me to have the particle list check if the colbox collided with any of the particles and return various info. Lame, I wish to change something with the AI class. I want it to inherit from playerControl. This might take some work, but it's definitely worth it as AI uses most of playerControl's functions. I'm just doubling up on code at the moment. Gonna figure that out tomorrow.

Also! I have also setup a SVN server for the project with various usernames and stuff. I plan to make a lot of changes on this project to make it more professional to gain a higher level of quality out of the end product. This means I really need to update the code and make supporting documentation for each bit of code. Need to add a new debug system too.

27th September – Today I changed playerControl to have more functions instead of really long ones. Then changed AI to inherit from playerControl and changed some of its functions to accommodate the changes. So, now the enemy AI can easily be hurt from hadouken attacks by the player.

Not sure what the next step is... Things I could do are: maker better debugging system OR make AI class work.

30th September – Not alot done since last time, just starting to pseudo out the AI class. Hard to do work on the game when I'm working full time.

5th October – Got the AI to decide whether or not to approach you, retreat, jump or do nothing at medium distance. Pretty easy to add in the rest of the AI from now. Also added a simple random number generator to the mix. Need to in the ability for the AI to attack, but that needs an attack table listing all attacks and what distance they need to be at to do them. Not hard, just effort at this point.

16th October – Haven't done any work recently, but on the weekend before I made a more efficient way for the AI to determine what to do. Now the AI can do all movement related parts for short long and mid distance. Next is attacks.

8th November – Finally got the ai to attack. It doesn't decide very well yet, but its a start. A few things to note is that:

!!! Attacks/Moves have to be 1 word and less then 80 characters. !!! Use underscores for multiple words !!! Attacks cant be called null/NULL
--

This is okay though. When a in game command list is displayed I have to remember to have it so that when names are displayed that underscores are taken out. Not now's problem. Just have to remember.

23rd November – Finished off the get attack code for the AI, but unable to test with ducking yet. Been ages since last update so not sure what else has changed. Jamie has been trying his hand at the music and I've done a little research into it. If I cant play multiple songs at once I may need to find another library for music alone. Next is blocking. Few more small bugs.

30th November – End of another month. Ai and player both know how to block now. Damage was fixed up so you can get hurt whilst ducking or what was considered otherwise 'busy'. Few more bugs but a new demo release is being made and will be on the site shortly, if not when you read this.

4th February 2010 – First post of this year, times getting on it seems. Looks like blocking is somewhat bugged.. Also, added bounce back and run forward. Now to add aerial bounce forward/back. I might need to start looking at bugs soon... List is getting really big.

24th February – Added aerial bounce forward/back. Cleaned up most particle related bugs and a few others. Let's see how many bugs are left...Say 12 left? It's hard to say,

looks like I wrote a couple of these bugs down a couple of times. Glad things are getting cleaner though.

7th March – Thank Christ for that. I was stuck on a bug for ages, since the last post. So when you took damage from attacks (mostly particles), there was a chance it would leave a rectangle after image of the previous animation before the attack animation. So the solution was (apparently) to do the clearBG function at the start of the animation switch. I put these damage collision checking functions into the draw function (I know that's the wrong place to put it... may move it, but) this ensures that all the drawing related stuff and anim changes happen at the same point. If it doesn't change anything I'll move the damage collision stuff back. Oh and for reference sake, I'll call this bug the "drag effect". There still seems to be a few more bugs left though. One bug is that on the far right side of the screen if your 'inside' the other player/fighter while they are punching, you end up facing the same way as them until they return to a standing animation. This tends to leave little 'drag' marks, but I think mostly these drag marks happen from running. Cant Say I know or care at the moment, later's problem. I feel like more development and that means I'm moving onto linked attacks or adding conditions to frames somehow.

23rd March – I have done butt loads of work since the last post, so I'm going to explain each bit in significant order. First of all, all drag effects should be fixed forever. Solution was to make only the program decide when to draw things, but when my guys changed animations they would clear a patch of the screen. This clearing of the screen patch should not be up to them. Now it all happens at the right place, this is all thanks to Adam Parkes's advice.

NOTE: After attacks, for some reason it reads the first frame again. So if the first frame has sound, it plays twice. Need to have sounds on the second frame onwards.

Secondly, I have added my brothers HPBar class to the game. Now you can see you and your enemies health bars. This makes it feel more like a fighter, however nothing happens when one players health is depleted. That has to be added later. It looks very happy though.

NOTE: Need to have if the bar is displaying its energy in reverse order, to use the flipped frame. Currently it does not.

Thirdly and this was the most painful, was making you and the enemy have the ability to cancel and link attacks. First, some explanation; Cancel attacks are attacks which come in during a current attack. This is to mostly add the ability to have a sword swing out, but it would look stupid if they retracted it then suddenly did a slash from where the sword swung out to. This make it more fluid. Link attacks are attacks you can use directly after specific attacks, but input during the previous attack. This might be a two punch combo where returning to the original standing position looks correct.

Why this was hard? Well I had to figure out how to make the player link and cancel attacks. That wasn't so bad. However making AI use it was very difficult. Turns out whenever the AI uses an attack they change to the BUSY state, which is good, but, I didn't know it was doing that haha. To round things up, they can do all these things now.

This will now be a part of a small demo for the website to showcase my progress so far.