

Animation Game Design using Multimedia and Computer Vision Techniques

Zhenghao Chen

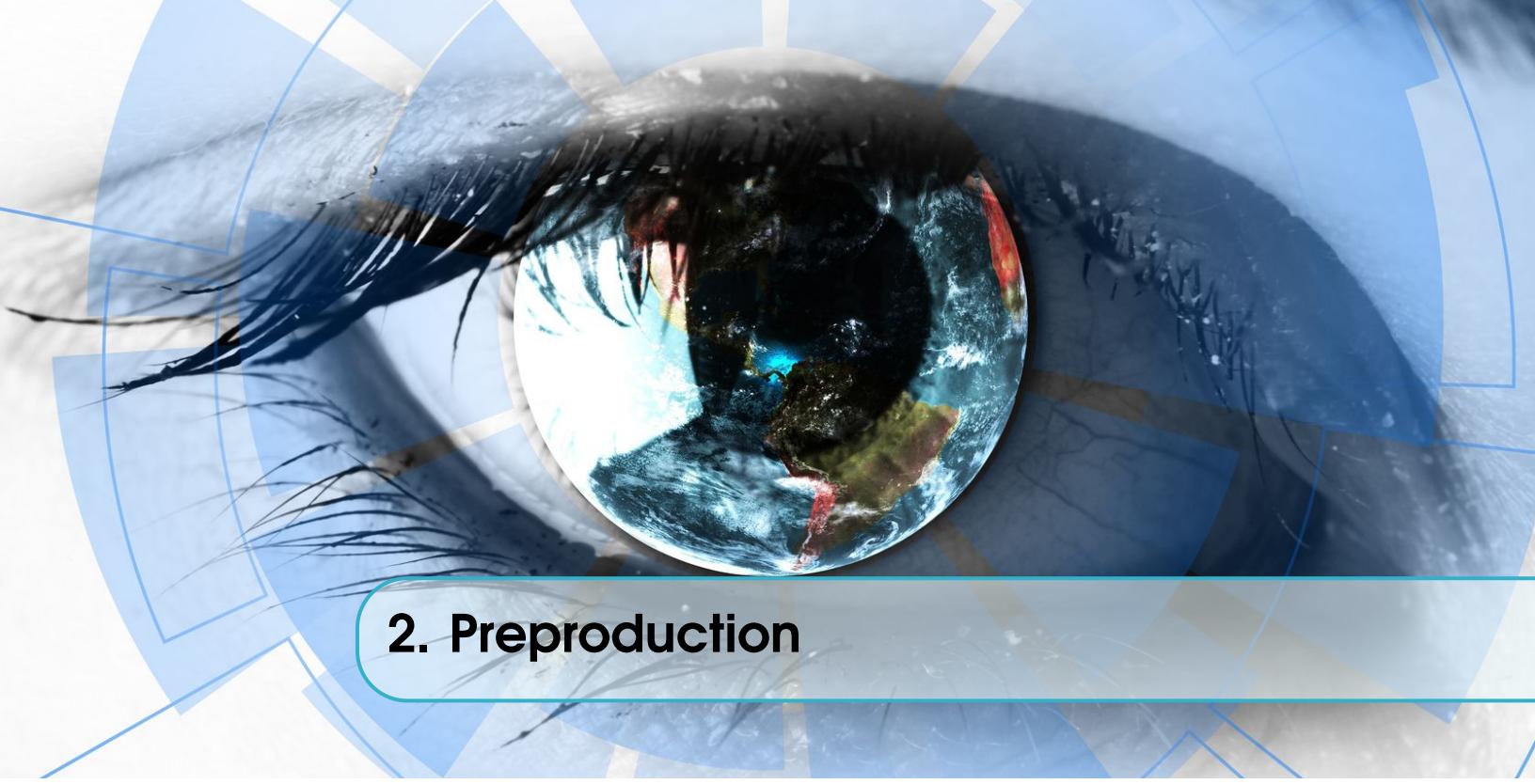


1. Abstract

This report introduces the design and implementation of animation game Protect White House. Specifically it will 3 main parts of the whole multimedia product that preproduction story line, character and environment design. Production that character modelling environment setting, sound track and animation control skills. Lastly, in the post production, some follow up like getting user experience and some associated production will be introduced.

In this project, the most significant contribution I did is that I created a new animation control skill that real-time motion tracking, capture and rendering. Specifically for tracking and capture I used a combined algorithm call max separation K-means combined the machine learning skills K-means and computer vision filtering skills K-means. For rendering, some typical image retouching skills are going to be implemented. Using this methodology, The motion control of the animation character will be simulated and created in real time from the real-world actor. This methodology is especially useful for low-budget animation created that the designer does not have to get motion capture equipment which is normally expensive. Instead capture the motion from the actor using this methodology is going to be easier.

Overall, this project is organized with well-design production, good implementation of the motion control. Excellent skills of image retouching and other multi-media skills.



2. Preproduction

2.1 Story Line

The main story line is that, the alien invades the earth and try to attack the white house, the play in the game has 2 missions to go which firstly is to shoot the alien using a flying object that controlled by user while the second task is to shoot the alien using a 3D machine gun. Once users complete these 2 games the media will be finished.

2.2 Character

Based on the story Line, there would be 4 different characters that would be designed for this media, that a Alien which its motion would be created by a techniques I will introduce in following section. As well as a Flying Object and 3D machine gun. In the game that 3D object and machine gun are going to controlled by users to attack the alien.

2.3 Environment

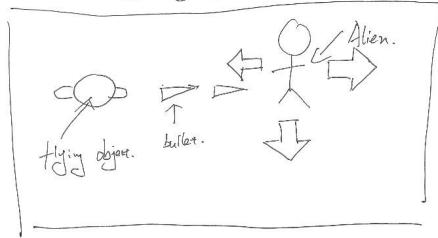
This game would have 2 main environments that for mission 1, it would be 2D environment while in mission 2 a 3D environment would be set up, as the game theme is about the alien attack white house, therefore both the environment should content white house to make sure the **Repetition** of design. Therefore the on-line photos I would like to use will be some white house relevant images .

Background:

1. Alien Attack the earth, try to blow up the white house.

Mission:

1. shoot the Alien down using constantly fly object. (2D-based).



2. shoot the Alien down using a 3D machine gun.

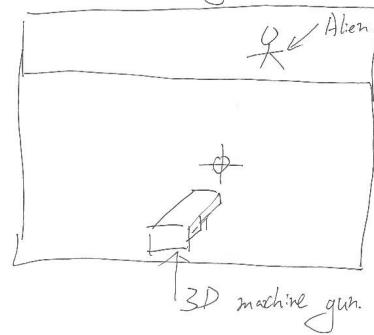


Figure 2.1: Story Board of Game



3. Production

3.1 Animation Character Motion Control

Simulating the motion of moving object Alien is the most essential stage of entire game which is captured from morphological operations of a my shooting video Moving-Monkey video. To achieve that, there are 3 main stages that are filtering, tracking and rendering.

3.1.1 Filtering

The five crucial parts of monkey body are labelled with red markers, that right hand, left hand, right foot, left foot and center of monkey. The first step to track the motion of monkey is to capture related pixels of these 5 points, in other words, we need to filter and remove other pixels. To do that a threshold will be set up for filtering. All pixels exceeds that value should be removed and the remaining pixels will be the parts of motions points of moving object. The algorithm is provided as followed.

```
for ∀ frame f in in video v do
    for ∀ pixels p in in frame f do
        if p > threshold then
            | remove p(set p to -1);
        end
    end
end
```

Algorithm 1: Algorithm 1 Filtering

This filtering method, the threshold I set up is red channel < 160 , blue channel > 120 and green channel > 120 .

Figure 2 shows the example of 1 frame that all 5 points of moving monkey are extracted using filtering method.



Figure 3.1: Example of filtering

3.1.2 Tracking

The next stage is to track the x, y coordinator of these 5 parts to locate the move and motion of the moving object, in this section, 2 algorithms will be introduced that are Max Separation [1] and K-Means [2][3]. Using these 2 algorithms we can track the motion of monkey.

K-means Clustering

K-Means is one of the most popular clustering algorithms. It stores k centroids that are used to define clusters. A point is considered to be in a particular cluster if it is closer to that cluster's centroid than any other centroids. Then it finds the best centroids by alternating between (1) assigning data points to clusters based on the current centroids (2) choosing centroids (points which are the center of a cluster) based on the current assignment of data points to clusters.

In our case, the distribution of selecting pixels can technically be formed in 5 clusters (left hand, right hand, left foot, right foot and center) that should correspondingly have 5 centroids (or 5 means) of all pixels considering each pixel is a 2-dimension vectors. The algorithm of K-means is as follows:

```

1. Initialize cluster centroids  $\mu_1, \mu_2 \dots \mu_n \subset \mathbb{R}$ 
2. Repeat until convergence :
for  $\forall i$  do
    |
     $C_i := \operatorname{argmin}_j ||x_i - \mu_i||$ 
end
for  $\forall j$  do
    |
     $\mu_j := \frac{\sum_{i=1}^{i=1} 1 \langle c_i = j \rangle X_i}{\sum_{m=1}^{i=1} 1 \langle c_i = j \rangle}$ 
end

```

Algorithm 2: Algorithm 2 K-means

However the K-means algorithm has very distinct limitation and tradeoff. For instance, assuming we use 5 means to cluster those selecting pixels, ideally we should have 5 centroids to represent left hand right hand, left foot, right foot and center. However somehow K-means computes only 1 centroid for both left foot and right foot as they are relatively close to each other therefore K-means consider they are belong to only 1 cluster.

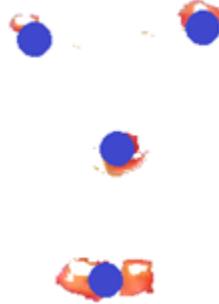


Figure 3.2: Example of tradeoff of K-means Algorithm

Figure 3.2 demonstrates the limitation of K-means, the 2 feet are supposed to be in 2 clusters but K-means consider it is one cluster. To overcome this problem, I use another techniques Max Separation.

Maximum Separation

Maximum Separation is a popular technique in computer vision. For tracking in this game, I also implement this technique to find the motion points. The main idea of max separation is find the maximum number of neighborhood of a pixel if we find there are a bunch of pixels located in certain area, that area could potentially be a motion point that we need to track. The algorithm is provided as followed.

```

number = 0
separate (pixel p):
for ∀neighbor n of p do
    if n ≠ -1 then
        | separate(n);
    end
    record coordinator of p
    number ++
    Remove p
end

```

Algorithm 3: Algorithm 3 Maximum Separation

Using algorithm 3 we can find the clusters of selecting motion points by maximizing the search area. On the other hand, this algorithm also has a very obvious weakness that is it is very sensitive to noise, which results in it sometimes mismatches the correct point.

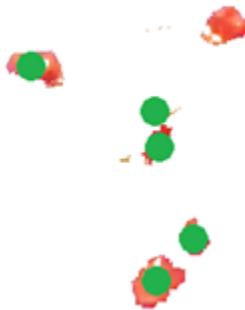


Figure 3.3: Example of Limitation of Maximum Separation

Figure 3.4 shows an instance of tradeoff using maximum separation that the right hand point is mismatch as one of points is targeted on the noise around the center point.

As a result, we see that both K-means and Maximum Separation have their limitations therefore it comes up a resolution that combines these 2 algorithms for tracking. The reason doing that is that results of experiments show that K-means performs really well for tracking the hands but poorly trace the feet while Maximum Separation can have an outstanding performance for tracing the feet

Figure 3.4 shows a really decent tracking result that combines these 2 methodologies, in this case, blue circles represent tracking results for K-means while green circles stand for Maximum Separation. About transforming the coordinators of those points to morphological operations of objects I will explain in following section, graphic rendering.

After we have those Motion Points, we can further connect those motion points using line connection and those lines are presenting that bones of the character that alien.

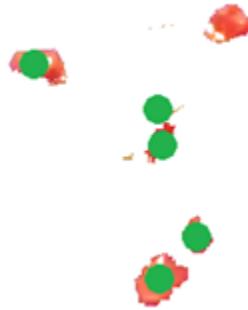


Figure 3.4: Combination of K-means and Maximum Separation for Tracking

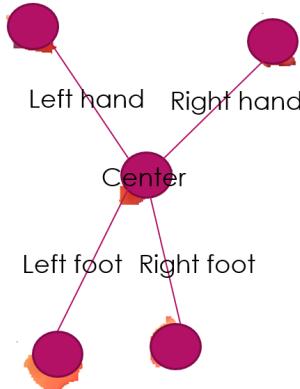


Figure 3.5: Bone of Character

3.1.3 Character Rendering

For the character rendering, I used **Image Retouching Techniques**. The source image used is a alien body with 2 arms and 2 feet. To adjust it to my character, firstly I **reset the region** the only cover the body without feet and arms. Then I set the **alpha channel** for the image make sure the image has transparency that is essential for compositing. Then I use **image compositing** techniques to add wings, and feet on alien based on the bones I have before. As I have the motion points and bones already the rendered character would be able to follow the motion the actor which is the monkey.

3.1.4 In-Game Character

Totally, I have 2 In-game character for one is UFO and for another is a 3D machine gun.

UFO is the first type of objects. The motion of that is controlled by mouse that x-coordinator and y-coordinator of UFO are obtained from mouseX and mouseY. The interaction of this object and moving object is that once this distance of UFO and Alienp is less than certain value I assume that alien hits the UFO and an explosion image will show to represent the crash of UFO. Also there are 3 UFO symbols on the top right representing the lives of UFO, every time Trump hits the UFO, it will

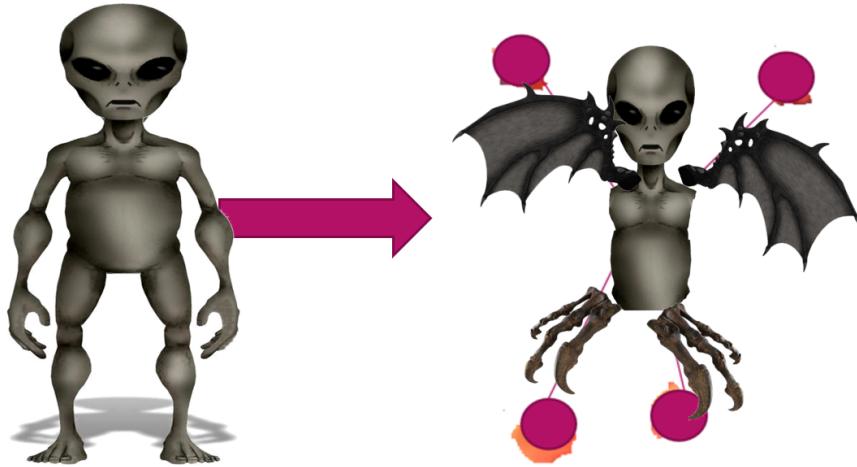


Figure 3.6: Image Rendering



Figure 3.7: Boom

drop one life.

The attack of UFO is controlled by mouse left click, once user click the mouse, the UFO will throw the weapon to the alien. There are 2 types of weapons. Firstly is **boom** is boom. This object can be created by mouse event left clicked when weapon option is Boom. The motion of Boom will be downward straightway affected by only simulated gravity. There are 2 different interactions of Boom. For one, if Boom hits the Alien that their distance is less than fixed value, the Boom will exploded while health value of Alien will be reduced for 10 points. For another, once the vertical position of Boom is greater than certain values, I assume that Boom is touched the ground so that it will explode. The second weapon for the game is **missile**. Similar to Boom it is created by mouse event left clicked when the weapon option is missile. There are 2 interactions between missile and moving Alien. For one, it can track the motion of Alien and attempt to attack it. Specifically, once missile is on left side of Alien, it will move rightward with randomly upward and downward to approach objective. In contrast, once it is on right side of Alien it goes left to attack the Alien. For another once it hits the object it will get explosion as well as reduce health point of object by 3 points.



Figure 3.8: UFO explosion



Figure 3.9: Missile

The next Character in the game is 3D machine gun, to make this 3D object, I follow an on-line tutorial[4], there are 4 basic types of 3D modelling techniques mainly used for this 3D object. Firstly, **extrusion** from a 2D plane is for the basic element. Secondly, the skill of **copy** is used for duplicating elements like bullets etc. thirdly, for further material, I have to set up diffused for material enabling the texture. Lastly, the light is setting up in the front of machine gun to make sure the **darkness** and **contrast** of this gun.

Also for the texture, I uses UV coordinator system as I want to export this object to be a **.obj** file, and UV system is good for coordinate the shape. The texture I download from on line source[5].

Overall, using blender to create this 3d Modelling make things much easier.

3.2 Environment Setting

3.2.1 Background

In the we need to set up the background for both 2D and 3D environment. As the game story is about protecting the White House, therefore the background would be a video about white house being attacked.



Figure 3.10: modelling of 3D machine Gun

To make this background video, first of all, I found a **still** picture of white house. Then I attach some other animations using **Processing Graphical library**. For each frame, it automatically generates the **image composites** and render as a video.

Figure 3.14 shows that for an example of a single frame, I attach the explosion effect to this image. By using tween animation skills I would be able to generate the video that used to be background video.

3.2.2 Sound Track

There are 2 types of sounds in the game for one is BGM that will keep playing during the game life time. Another one is interaction sounds that would only occurs when some special condition happens. In my game the media would play the explosion sound track once the weapon is touching the target which is the alien.

To implement the sound track, I simply use Java Beads[6] library which is currently one of the best audio libraries.

3.2.3 Game Pipe Line

Figure 4.3 shows the whole game pipeline, User try to kill alien in the limit time either users let Alien attacks 3 times or run out of time users lose.

Figure 3.11: UFO explosion

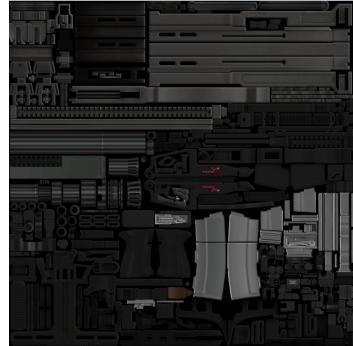


Figure 3.12: texture of 3D machine Gun[5]



Figure 3.13: 3D machine Gun in game

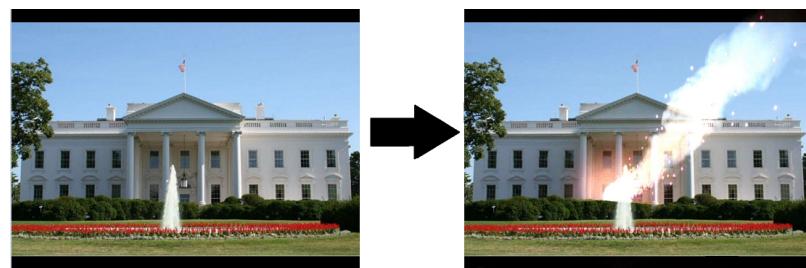


Figure 3.14: 3D machine Gun in game

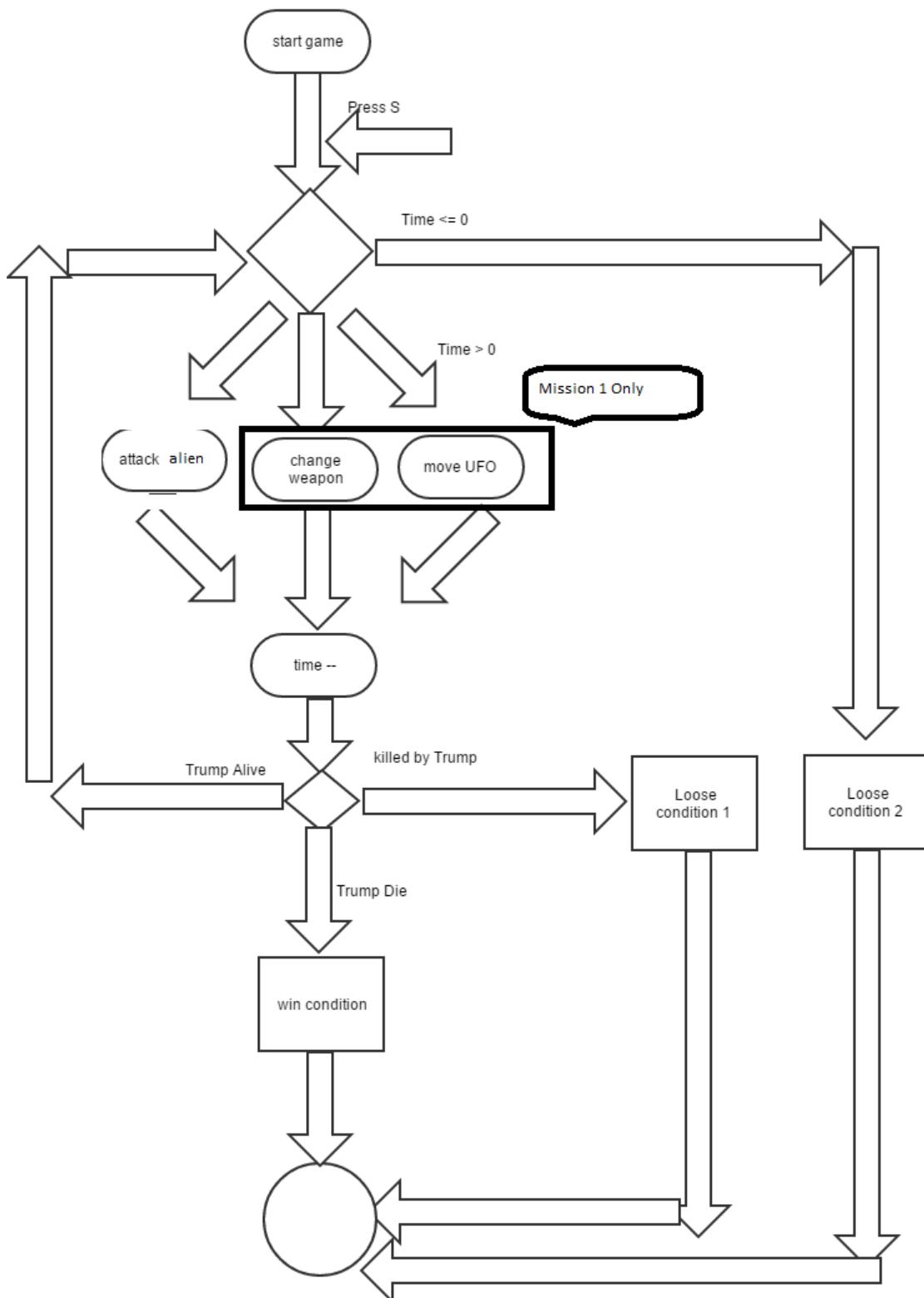


Figure 3.15: Game Pipe Line



4. Post Production

4.1 Associated Product

Post Production is also very essential stage. To promote my product and complete it, I basically did 4 things, Firstly, I created a logo for my game using 3D text skills and 3D animation by **Blender**, for this title I used **particle skills and lighting** to hight and get contrast for the words.



Figure 4.1: Logo of Game

The next thing I did is to create a poster, for my poster, I use **Image enhancement skills** including Gaussian Filter, smoothing edge, channel change, convolution filter etc.

the third things I did is that I created a trailer using video compositing skills, the trailer demonstrates the background story, the overall and basic story line for the whole game.



Figure 4.2: Poster

Lastly create a website using **HTML5, P5js and Jquery**, In that website, I have 3 things to show that for one, the background story which is my trailer, for two the character information, for three, I attaching the game demonstration.

In the home page Figure 4.4, players are allowed to play the bgm that star war track and navigate different function.

Specially, In the character page can you see the detail information for each character. In background page, you can see the background story including the trailer. In Game Demon Page, I include demonstrating video to show the how can look like in different missions.

4.2 HCI getting User feed back

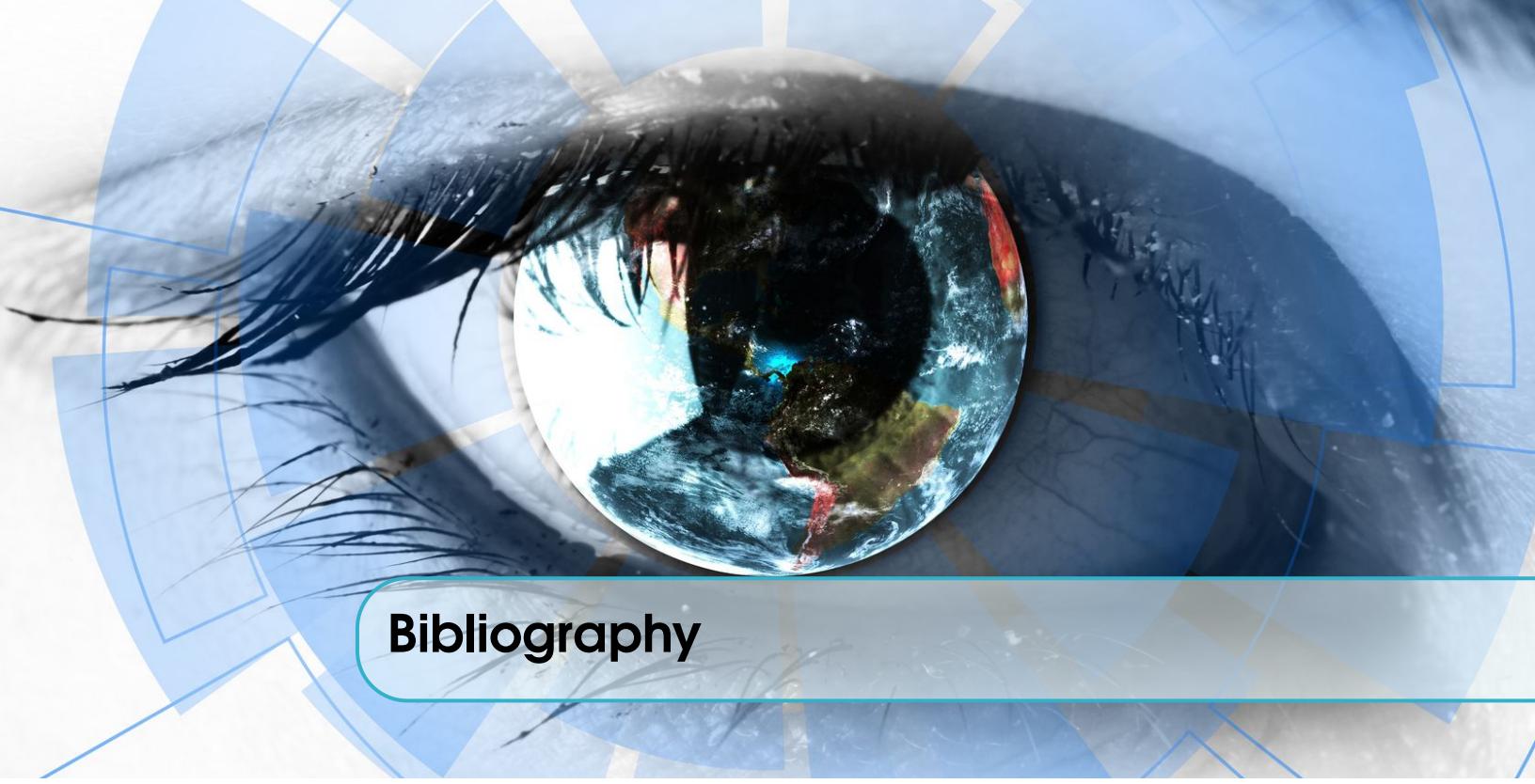
After design finish, I get the feedback from CJ, Hui Cui, Dr Xiuying Wang and other players to enhance the entire game quality.



Figure 4.3: Trailer



Figure 4.4: Homepage



Bibliography

- [1] Seferoglu, H. and Modiano, E. (2013). Diff-Max: Separation of routing and scheduling in backpressure-based wireless networks. 2013 Proceedings IEEE INFOCOM
- [2] Sites.google.com. (2016). k-means clustering algorithm - Data Clustering Algorithms. [online] Available at: <https://sites.google.com/site/dataclusteringalgorithms/k-means-clustering-algorithm> [Accessed 21 May 2016].
- [3] Stanford.edu. (2016). CS221. [online] Available at: <http://stanford.edu/cpiech/cs221/handouts/kmeans.html> [Accessed 21 May 2016].
- [4] 3D machine Gun tutorial with blender <https://www.youtube.com/watch?v=ovPsVhSP4xQ>
- [5] Texture Source of CZ805 <http://callofduty.wikia.com/wiki/Vector>
- [6] Beads in Java <http://www.beadsproject.net/>