Nibok

A method for create and develop computer games

Gholamreza Sharif Zadeh
(Emperor Katax)

In Memory of Psygnosis.

In principle a work of art has always been reproducible. Man-made artifacts could always be imitated by men. Replicas were made by pupils in practice of their craft, by masters for diffusing their works, and, finally, by third parties in the pursuit of gain. Mechanical reproduction of a work of art, however, represents something new. Historically, it advanced intermittently and in leaps at long intervals, but with accelerated intensity. The Greeks knew only two procedures of technically reproducing works of art: founding and stamping. Bronzes, terra cottas, and coins were the only art works which they could produce in quantity. All others were unique and could not be mechanically reproduced. With the woodcut graphic art became mechanically reproducible for the first time, long before script became reproducible by print. The enormous changes which printing, the mechanical reproduction of writing, has brought about in literature are a familiar story. However, within the phenomenon which we are here examining from the perspective of world history, print is merely a special, though particularly important, case. During the Middle Ages engraving and etching were added to the woodcut; at the beginning of the nineteenth century lithography made its appearance. With lithography the technique of reproduction reached an essentially new stage. This much more direct process was distinguished by the tracing of the design on a stone rather than its incision on a block of wood or its etching on a copperplate and permitted graphic art for the first time to put its products on the market, not only in large numbers as hitherto, but also in daily changing forms. Lithography enabled graphic art to illustrate everyday life, and it began to keep pace with printing. But only a few decades after its invention, lithography was surpassed by photography. For the first time in the process of pictorial reproduction, photography freed the hand of the most important artistic functions which henceforth devolved only upon the eye looking into a lens. Since the eye perceives more swiftly than the hand can draw, the process of pictorial reproduction was accelerated so enormously that it could keep pace with speech. A film operator shooting a scene in the studio captures the images at the speed of an actor’s speech. Just as lithography virtually implied the illustrated newspaper, so did photography foreshadow the sound film. The technical reproduction of sound was tackled at the end of the last century. These convergent endeavors made predictable a situation which Paul Valery pointed up in this sentence:

> “Just as water, gas, and electricity are brought into our houses from far off to satisfy our needs in response to a minimal effort, so we shall be supplied with visual or auditory images, which will appear and disappear at a simple movement of the hand, hardly more than a sign.”

Around 1900 technical reproduction had reached a standard that not only permitted it to reproduce all transmitted works of art and thus to cause the most profound change in their impact upon the public; it also had captured a place of its own among the artistic processes. For the study of this standard nothing is more revealing than the nature of the repercussions that these two different manifestations – the reproduction of works of art and the art of the film – have had on art in its traditional form.

The Work of Art in the Age of Mechanical Reproduction - Walter Benjamin (1936)

Source: UCLA School of Theater, Film and Television

Transcribed: by Andy Blunden 1998; proofed and corrected Feb. 2005

Web address:

http://www.marxists.org/reference/subject/philosophy/works/ge/benjamin.htm

Table of Contents

Chapter One - First steps - D level

Introduction

Game production is one of the most creative and technical topics in the modern world. It supports big business and also it is young. In this note, I describe a method for creating computer games. I try to use all possible “Human Resources” for create, design and product computer game. It means, every section of this method can be provided by numbers of expertise and experiences of mankind in Art, Technology, Economy, Journal, History, Computer and…

A computer game is an interactive environment which an artificial entity can understand it and use it. Playing game means putting time and concentration for analysing some data and prepare some respond for them during periods of time. It can supports some movements between players. It can be mentioned for some special moments. It can be sold in millions.

This note is focused on creating high quality games by using **Nibok** method.

Method for creating computer games

The **Nibok** is a method for creating computer games, by supporting all possible/impossible details and complexity in various genders. This method is not depends on hardware or software but works with "Human" factors, because of that "**Nibok**" can be used in all environments, cultures, region and level of game design technology.

Nibok method started with these steps:

1. **Create logical graph of game**

   The fact is, from start to end of the game what is our "Movement", and player should obey which kinds of rules and structures in logical way. This logic could be linear or based on a math's graph. It can use logical circles, special situation and logical gaps. The game logic graph is a graphical presentation that shows the game logic in a brief presentation. It is like a flowchart but more flexible.
   For drawing a simple logical graph first assign 2 points for your game. Point "A" for start and point "Z" for end. Place these points in a circle. (see figure S1-1)

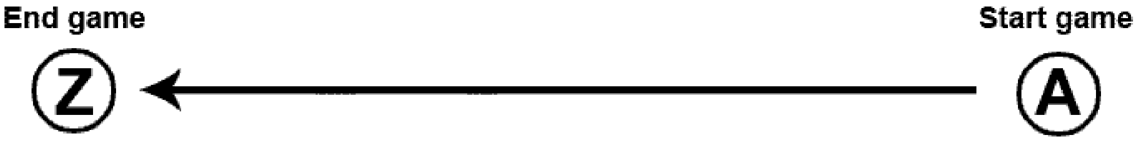


Figure S1-1 – Simple move direction

Then place an arrow to show the "Move Direction" of movement between point A and point Z. Now we have one game logical graph. It says that players starts from A and then move to Z.
We can have more than one arrow from "A" and "Z". Each of them presents a different "Move Direction". In figure S1-1 player start from "A" and finish game on "Z". This is a simple move direction. But in figure S1-2, player can start from "A", then finish at "Z" and then move back to "A" and follow the game again and again. This is new game logical graph that is difference from previous one.

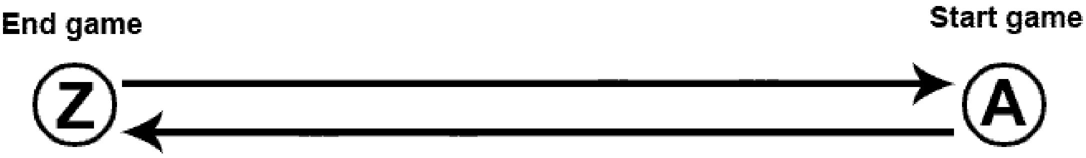


Figure S1-2

Many simple games and some network games used logics of figure S1-2. Adventures games and puzzle games follow the same logic. The figure S1-3 shows the logical graph that used in these games. They have many steps that related to each other by one or more "Move Direction".

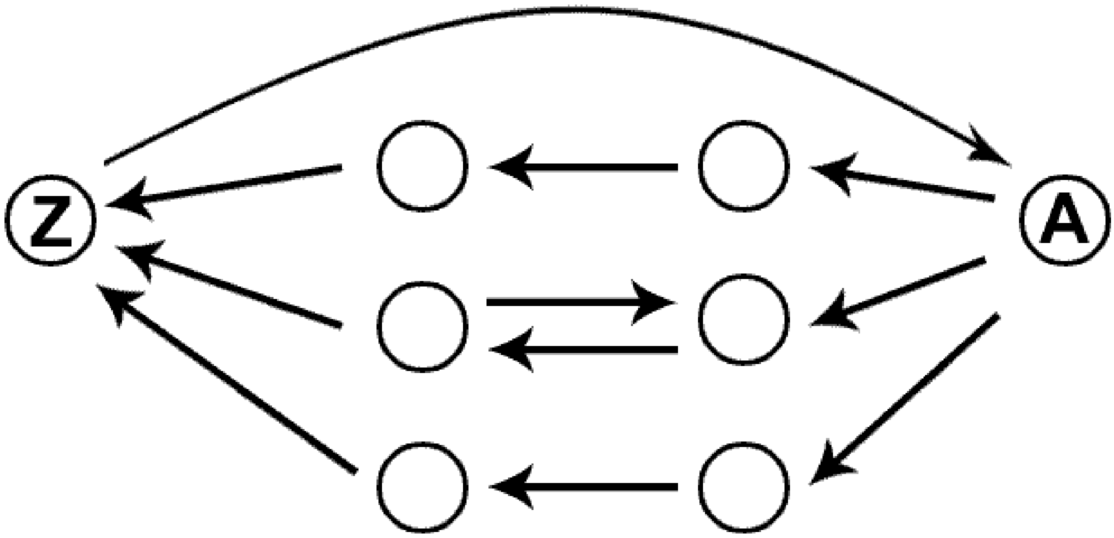


Figure S1-3

Sometimes players can “Create” a new destination based on past experiences during playing game. This logic represent on figure S1-4. As it shows, the first end of game is “Z” but players can follow the game and reach to a new “End” that presents by “ Z` ”. Many games that are based on economy and genetic algorithms are following this logical graph.

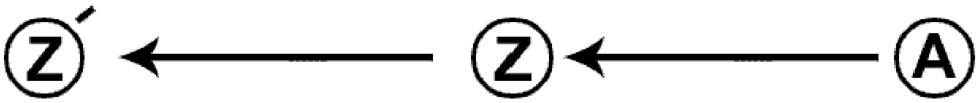


Figure S1-4

Let’s have another look to the previews logical graphs. Each of them has 2 circles with “A” and “Z” that presents the start level and end level of game, there are more circles that present other levels during the game story there are some arrows that present move direction between circles. So the question is how player play the game based on these logical graphs? The answer is, player starts from “A” and finish on “Z”. Every arrow from A to Z represents one rule for reaching to Z. For create a real concept of each circle, we use “Information and Data”. For creating a real concept of each arrow, we use “Connection and Relation”. In other word, each **circle** is made by information and named **information unit**. Each **arrow** is made by rule and known as **rule unit**.

Look to these examples:

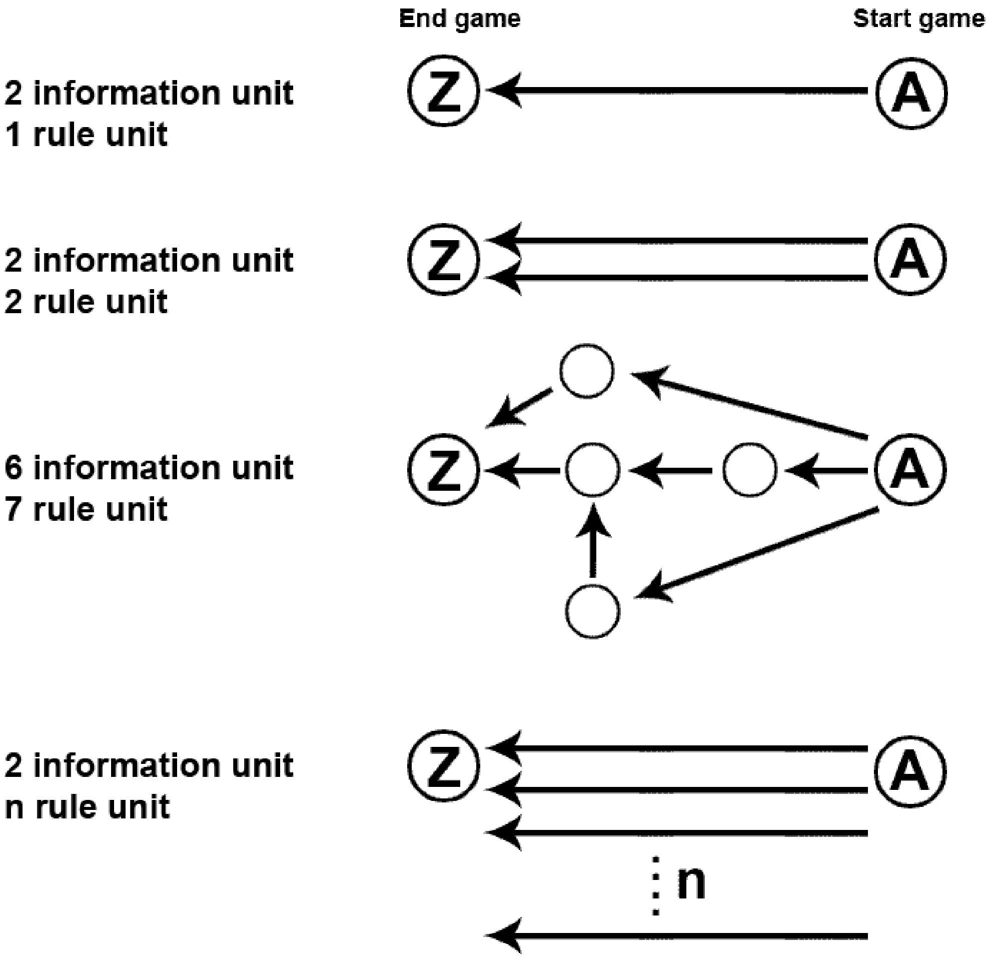


Figure S1-5
I use “I” for representing numbers of information units and use “R” for representing numbers of rule units between information units. “R” and “I” are named as “Game Units”. These are two main elements for designing game based on “**Nibok**” method. This is a formula based on this information:

I + R = T

“T” represents **total units** for design, program and create one game based numbers of information units and rule units. “T” on the other hand presents an estimate time for creating a game. I will talk more about “T” in future. Look to these example:

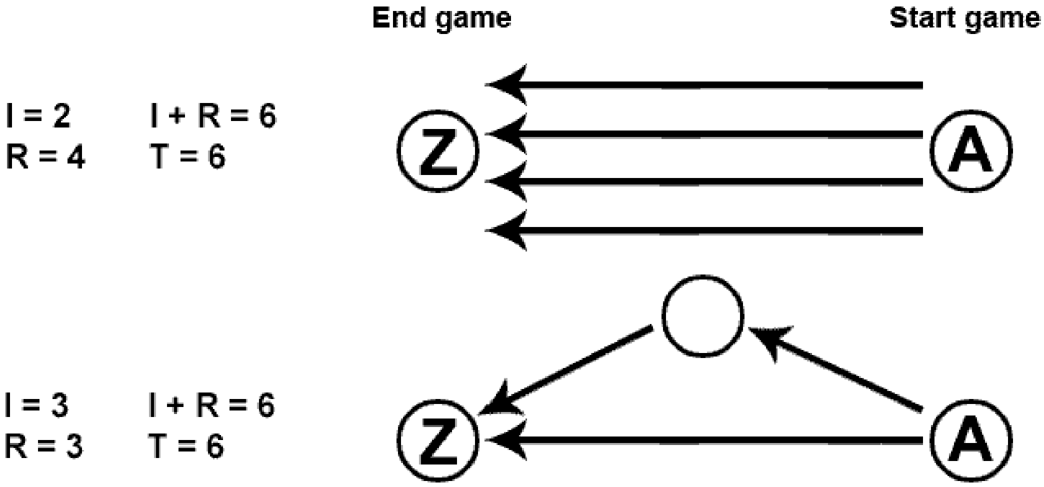


Figure S1-6
As you see for each of these logical graphs, “T” is equal to 6. But the units on logical graphs are different. It shows there is an equal value for complexity as logic or as visual/audio items related to game is possible. This equality is important when you have lots of information units and rule units. All of these units, as you will see, must be prepared in high quality details for a step by step development. “T” is not presents time but can supports and describes amounts of works that must to be doing for creating game. I will talk more about this on next pages.

2. **Prepare game story based on Logical graph**
   After drawing the logical graph of a game, now we can **stabilize/write/convert/create** a story to this graph. There are thousands points of views that can be used in this era. Artistic view, psychological view, emotion and drama or creative and modern concepts. How long is the duration of the game? Who are the players and how much expertise is needed for the game? These are questions that present themselves in this area. I prepare some examples that show how to **stabilize/write/convert/create** a story based on specific logical graphs. These are important simple examples and I will refer to them in other sections of this note.

Example One:

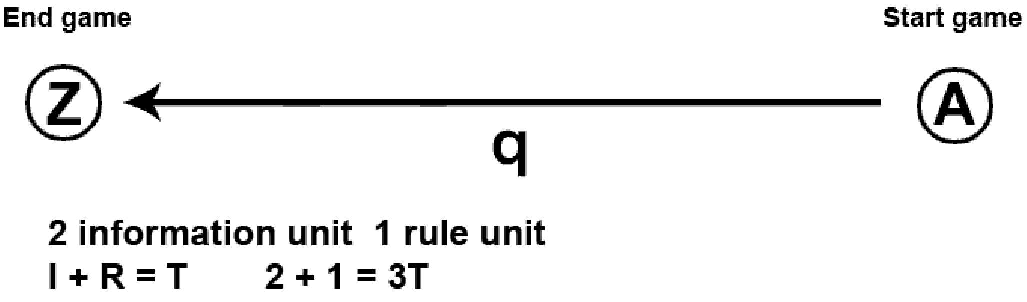


Figure S1-07

Story: Player is an archer. He must shoot at some apples. When he targets all the apples, game will end.
Description:
A: This is the main stage of game.
q: This is a name for a function which controls shooting an arrows. It could be a direct shoot to target (apple) or in-direct shoot to target. (see figure S1-8)

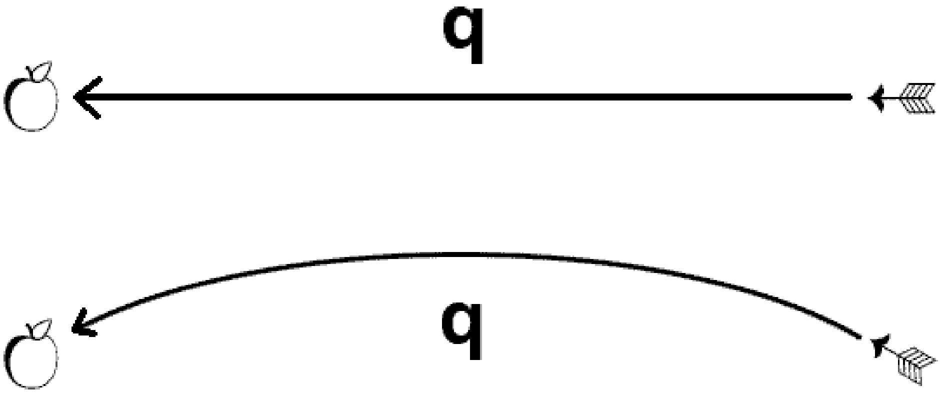


Figure S1-8
Z: This is an ending stage of game. It can be an image for records of high marks or a special movie.

Example Two:

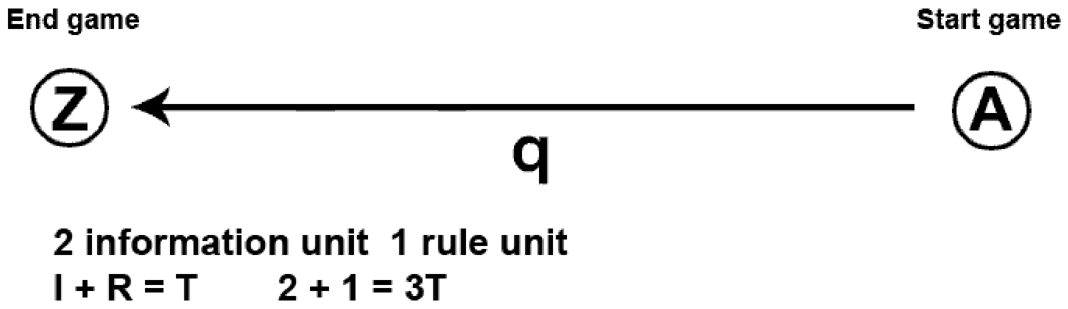


Figure S1-9

Story: A small grey squirrel is passing from a dangerous mountain and fights with some bad animals and reaches to his house.

Description:

A: First place of small grey squirrel.

q: Methods of move and fight for small grey squirrel with enemies.

Z: The home and ending of game.

Example Three:

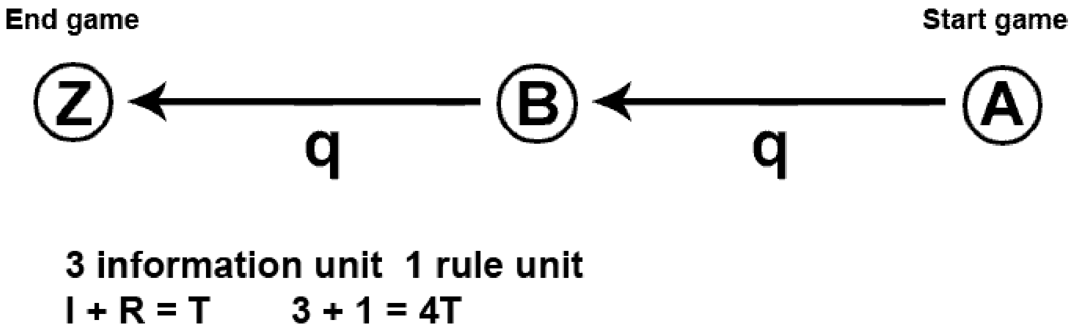


Figure S1-10

Story: A space ship move throw a path to reach to it’s headquarter. It fights with enemies during the game.

Description:

A: Start point of game with selecting various models of ships and weapons.

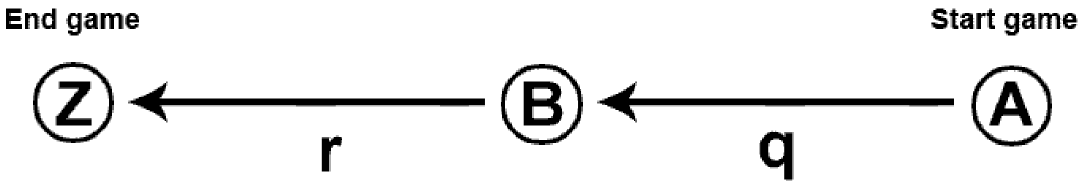
q: Methods of attacks and destroying enemies.

B: Environment that space ship fights inside that.

Z: Headquarter

Note: “**B**” is an information unit that describe an environment of fight and attack. Before and after fight, all methods for attack and fight (**q**) are same. Have a look to next example for understanding the differences.

Example Four:



3 information unit 2 rule unit
I + R = T 3 + 2 = 5T

Figure S1-11

Story: A space ship leaves mother ship for bombing a target. It fights with enemies during the way. After bombing he must reach to headquarter, and fighting with much more expert enemies.

Description:
A: Mother ship with full arsenal for space ship.
B: Fight area
q: Methods of moving space ship with bomb, and fighting with enemies.
r: Methods of moving space ship after dropping bomb. It moves faster. Enemy units also have more shooting accessories.
Z: Headquarter

Note: Based on "**Nibok**" method, first we draw logical graph of game and then we adapted the real story based on this graph. We can add more detail to the graph if the story needs more. Adaptation between story and logical graph guarantied the speed of creating game on the next steps of **Nibok** methods, so the story and the logical graph must cover each other perfectly. This prepares a transparency between real story and digital version of that (computer game).

Example Five:

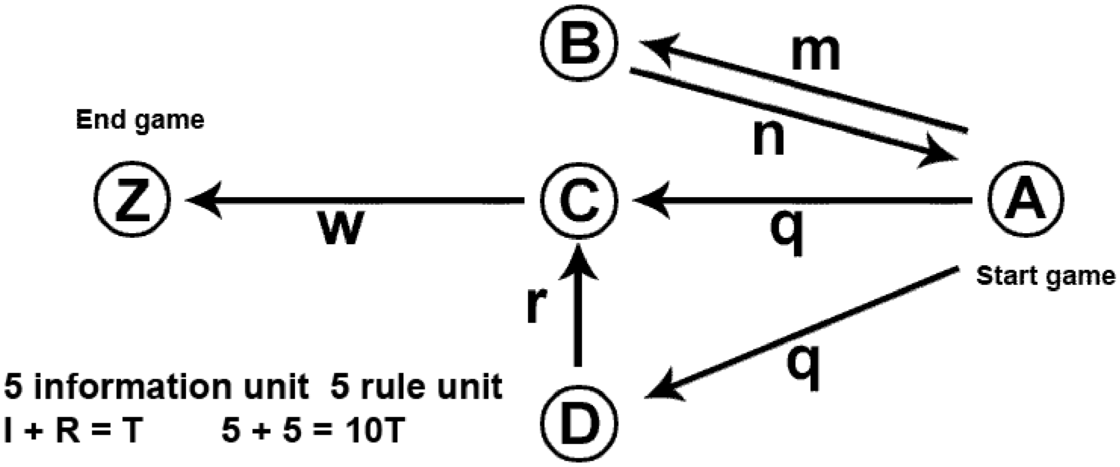


Figure S1-12

Story: A boy in a dream land wants to bring for his little sister some cacao and jam cockies. For buying these items he must collect golden starts, and then give golden starts to kind old mans that prepare the cookies and get the cookies. The golden starts are in hands of a bad witch. Their shapes are circular and sharp and both of them are ok for getting cookies.

Description:

A: Start the game in a dream land. There is a little girl who is sad.

B: Castle of bad witch.

m: Methods of moving to the castle and fighting with enemies for catching golden stars.

n: Methods of moving back from the castle with golden stars and fighting with enemies.

C: The kind old man`s cooking shop which accept golden stars and give cookies.

D: The kind old man`s cooking shop which accept golden stars and give cookies two times more.

q: Methods of moving to cooking shop through some traps that created by bad witch.

r: If boy buys cookies in **D** then there is a special enemy to attack him.

w: Methods of moving to home, Here the game difficulty is high.

Z: End of game and a happy little girl.

- If player set game difficulty to “Hard” then “**r**” will assign to game. In “Easy” mode it should be replaced by “**q**”.

Note: The next example has the same logical graph but the story is different.

Example Six:

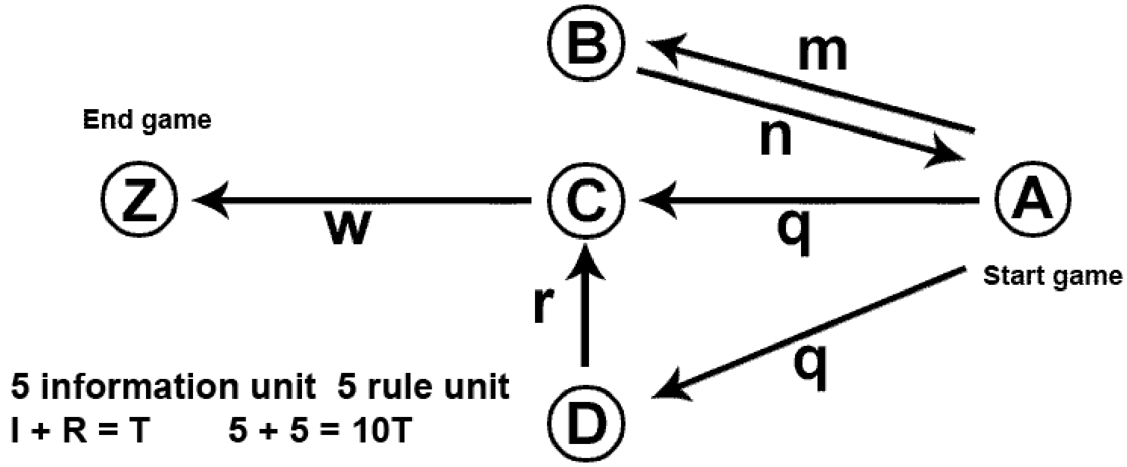
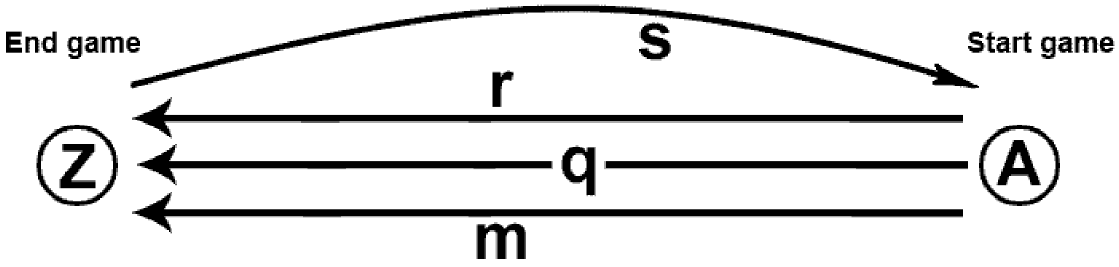


Figure S1-13

Story: A business company which sell farm product, need more production to bye modern farming tools. At the beginning of game you have 10 hectares of farm and $1000 credits. You must produced more product and increase your credits. Then bye modern farming tools and increases your farming production. The goal is reaching to $100,000 credits.
A: Small farm with $1000 credits.
B: Farm with separator grain factory for farm's products.
m: Farming methods
n: Farming methods by using factory
q: Sending products to shopping centre by normal transporting system.
D: Shopping centre A: Local and small
C: Shopping centre B: Far and big
r: If there is not a good business in shopping centre A, we can send products to shopping centre B. Other farmer's business and natural disasters can affect the shopping centre A.
w: Upgrade transporting tools and farming tools
Z: End game with collecting $100,000.

Note: Both of these examples have equal information and rule units and **T** is equal to 10 for each of them, but the story and style is totally different. The logical graph of game is like bones of game's structure. We can apply and create various kinds of stories based on this structure. Because all of them have same **T** = 10, then for creating the game there is a specific "Time and Energy".

Example Seven:



2 information unit 4 rule unit
I + R = T 2 + 4 = 6T

Figure S1- 14

Story: You navigate a Boeing 747 between two airports. This is a simulation game and contains different weather situation as level of difficulty for players.

Description:

A: Starting airport

Z: Destination airport

q: Normal weather's methods

r: Special weather's method

m: in this situation, your plane has some problem in navigation.

s: The pilot is ranked and he can play game continually for earn higher ranks and increase his expertise.

Note: **A** & **Z** can be random locations but their information is absolutely same. Methods for **q**, **r** and **m** could be dynamically changed during the game period based on player experiences.

3. Description of units

Now we move one step deeper into this method by describing "**information unit**" and "**rule units**" in more details. Here I describe what sort of things is related to these units and how to prepare them in useful standards. Now we can gather big amounts of information from people who are expert on psychology, air navigation, economy, military, journalists, story writers .. and use their knowledge and experiences to create a game. Even we can adapt a real story on a logical graph with saving levels of story details .The "**I + R = T**" formula represents the amounts of information and data that must be prepare for a game. We can estimate a period of time based on **T** for designing a game. On the other hand, **T** presents the “**complexity”** of a game. Here is an example related to describe **complexity** of game:

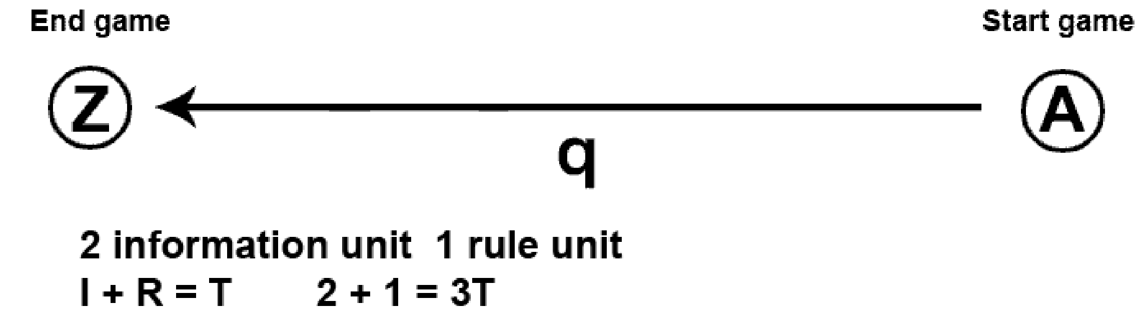


Figure S1-15

Story: An atomic explosion is occurred in a city. You must be surviving by using some medical kit during a specific period of time.

A: City before explosion

Z: City after explosion

q: Destruction methods

Here I need 3T for design but if the logical graph be like figure S1-16, I have a different value for T:

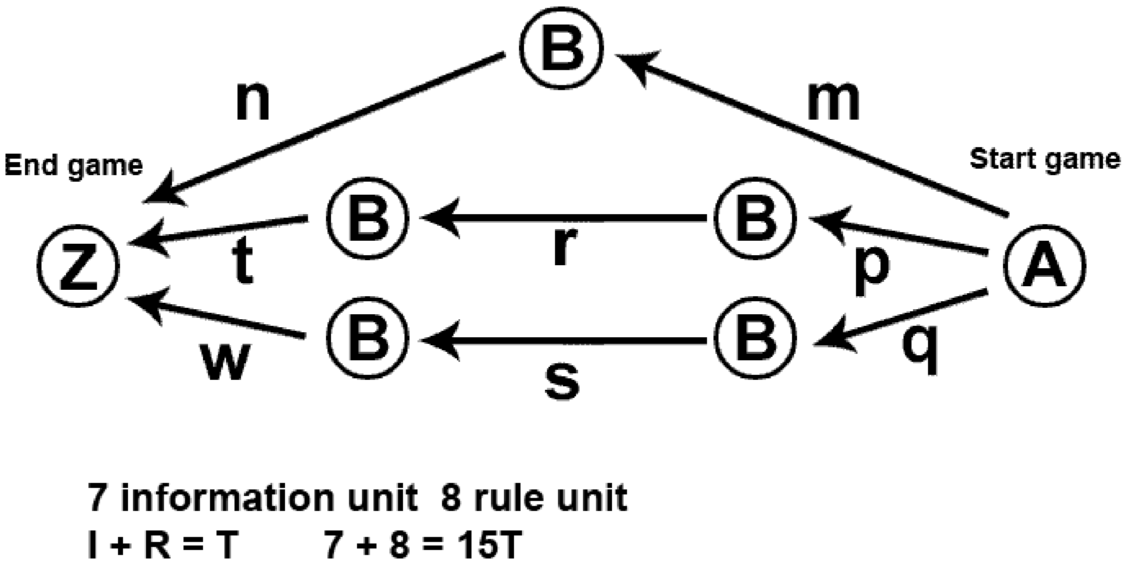


Figure S1-16

Story: An atomic explosion is occurred in a city. You must be surviving by using some medical kit during a specific period of time.

A: City before explosion

p: Radiation effects on plants   **q**: Radiation effects on animals

B: First waves of explosion’s shock on alive entities

C: First waves of explosion’s shock over environment

r: More radiation effects after first waves of explosion’s shock on alive entities depends on distance

s: More radiation effects after first waves of explosion’s shock on environment depends on distance

D: Peeks of destruction (alive entities)

E: Peeks of destruction (environment)

t: Final destruction effect on alive entities (Fear and pain)

w: Final destruction effect on environment (Pollution and radiation)

F: Distance visual/Audio effects of explosion

m: First waves of explosion’s shock on distances

n: Final waves of explosion’s shock on distances

Z: Survivors

For design a game based on this logical graph we use T = 15 time unit that in comparison to last one is 5 times bigger. This needs more energy for collecting resources and more time to create units and the reason is more detailed game based on story.

Whit calculating **T** value, before preparing a game, at the first steps of project we can have a schema of “How long” and “How much” a game need time and money for design and create. This can prepare a good business plan for next steps.

Now I describe “**information unit**” and “**rule unit**” in details. They have their own species, methods and providers. For designing each of them, we can use lots of “Human resources”. It means everybody, apart from level of knowledge on computer science and game industry, can join to create a game story and game items. This is one of the benefits of **Nibok** method. This method is using all possible expertise and information for create a game.

Information Units

What is an “**information Unit**”? The **information units** are the areas that players are playing game inside them. They are stages that variables and characters are engaging in for playing game. They contains game’s concept. After player finish a game, all he/she remembers are these information units. They can be used for next version of game. The long term effects of game are prepared by these units. They are designing by creator of game based on story, but before describe how to design them I want to follow players on visiting an information unit.

Player passes these steps after visiting an information unit:

1 - **Watch and behold**: Watching the environment and memorize details.

2 – **Thinking and compare**: Tools, numbers, icons, colours, lights and sounds of an environment prepare different response in each player. They arise questions or represents memories of the past.

3 - **Move and search**: Gathering data from an information unit can be provides by opening, reading, listening, turn on or turn off, use power or limited power, fast or slow-motion movement and... It is like Rubik's Cube that player first start to looking at that from different angles.

4 - **Random effects**: Player can be targeted by some random effects like thunder’s sound, appearing or disappearing some items, a special text or sign or applying an animal’s motion to a tool or entities. These effects can be prepared by information unit as random or sequential ways of presentation.

5 - **Understanding and memorising**: Player reaches to this step after last 4 step. Here he/she analyse the information unit and memorise it. Then he decides to play the game and discover/learn the rules of playing that game.

Based on these 5 steps I prepare a list of important features for designing an information unit:

1- **History**: Materials and historical concepts of each icon, graphic and sounds

2- **Sex**: The gender of environment, is that a presentation of male hood or female hood or other unknown entity or a mix of all. What is the fashion mode of environment? Is that representing a woman or a man?

3- **Light**: Colours, lights, fade areas and bright areas, quality of colour and light, blur effects.

4- **Date** and place: Time and place of game. It can be performs by visual and audio elements.

5- **Art and feeling**: What is the feeling of game? Is it “Kind” or “Brutal”? Is it “Warm” or “Cold”? Have movement or it is motionless?

6- **Data**: Choose how many language we use in our environment and which sorts of data must use that specific language. Is that data “Clear” or “Mysterious” or “Psychological”?

7- **Important characters and icons**: In many games, there is one or more character or icon that is in centre of attention. Are these characters or icons, “Hidden” or “Visible”? Are them appearing randomly or player must find them?

Now I compare two of previous examples to understand how to design information units in action:

Example 5 – information unit B – “Castle of the Wizard”

As the story shows, boy is looking for golden stars and he reaches to “Castel of the Wizard”. He must fight there and conquer that castle. Here are the details for this

- Colures are in fade style with grey theme.
- Sounds have high volume and echoes in silence.
- All random actions have horror theme.
- Castle’s walls have a large range of magic icons.
- Scales are big and vast.
- The Wizard’s location is hidden but could be discover by some icons and light effect.
- The golden stars are behind some doors. Paths for reaching to these doors are mostly indirect.

Now imagine a player is looking to this featured scene and suddenly he turns off computer, answers a call and moves outside of his home. The things that he remembers from the game are based on these items for this game. These items are related to this game, not other games. So the quality of concepts and styles of designing this information unit is directly affect the player’s attention to a game and create a common memory for this specific information unit between players efficiently. Look to the next example.

Example 7 – information unit A – “Starting airport”

- Vast view for a big airport.
- Day and Night effects
- Quality of air-control tower’s sounds as male & female and English accent.
- Co-pilot nationality and his/her behaviour
- Ambient sound of airport
- Design of the pilot’s place in cases of “Have more details” or “All details MUST be visible”.
- Data presentation on air plan navigations (analogue or digital).

As you see, an information unit is a settle point on a computer game. It is a stage which all programmed data and reasons are presented for player. So design this unit covers many aspects related to design but, because game is an interactive product, in creation process of an information unit, we must be aware of two main line for transferring data between computer and human’s (player) brain. These rules arise from interactivity and naturally create static and dynamic concepts for game:

Rule A- Data is gathering from game by player’s brain and saves as unique unit in his brain. For example a special sound or colour that represents danger. Player save this sound or colour in his mind and when he found it again, he understands that there is danger in that location. All these things are indicated in information unit. This is a static concept in designing information unit.

Rule B- Data is zipped and packed in game, when it analysed in player`s brain, it opens and shows many edges of possibilities or new things related to that stage. For example players visit a special icon or an unusual shape in game. This special icon or shape can generate many related data to main story in player’s mind and push him/her thinking about them more deeply. This is the most effective way for designing an information unit. Sometimes it (information unit) can design as short accident in game. Sometimes that (information unit) is very big and highly compressed and players suddenly found it and his mind generate bigger parts of data related to that in his brain. This caused high energy in player and make game more exited. Sometimes it is a sound or special frequency for move. This section is one of the most beautiful issues in creating information unit for a game. It shows interactivity of game more alive. This is a dynamic concept in designing information unit.

For example in game “Tetris”, for designing the environment of game and shape of details, **Rule A** can be used. For appearing next pieces in game, **Rule B** can be used wisely. It makes game more tensions. This is logical graph of game “Tetris”:

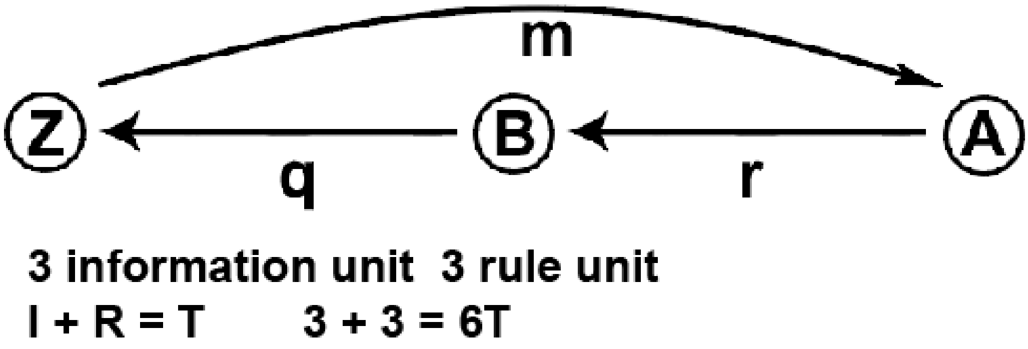


Figure S1-17

Story: There are cubic pieces which follow from the top to down of main screen. Player can move and rotate them. If player can put them in a line, then he has score and the lined cubes will disappeared.

A: Game stage with shapes and pieces

Z: End of game by disappearing lined cubes.

r: Calculation of movement for first piece by speed and type and move it to stage.

q: Changing the rotation and position of pieces by player

B: Next piece shows before come to the stage

m: Move to next level of game by calculating score and level of difficulty

By focusing on information unit B, you can analyse it in many ways. This directly affects your game and players. There are many versions of “Tetris” game but only few of them are in player’s favourites, because of wise design of information unit B.

Analyse of information unit

Main point in analysing an information unit is creating lists of elements for designing this unit. After reviewing whole the logical graph and other related information to game story, now the time is for create listed items that will be used on the next level of creating game. I named these lists, “**Theme List**”, “**Item list**”,” **Environment list**” and “**Prodigy list**”.

1. Theme List - Background of information unit and its properties

Every information unit has a background. Background of information unit is the “Place” or “Stage” that the game is playing there by player’s commands. This “Place” could be like “Plate” or “Space” or even “Single Colour Screen”. Background has some properties that maintain background features. Background properties can be simple or changeable. If there are multiple changes on background, it must be listed. The ambient sounds on background are other items that must be listed. For example, volume and frequency and quality are some properties related to ambient sounds in background. **Theme list** is used to describe all these features.

Example Two – information unit A - First place of small grey squirrel

> **Theme list**: There is a big wild jungle in summer, with moving clouds and bright shadows of sun on far mountains. Ambient sound is natural sound of jungle.

Example Three – information unit B - Environment that space ship fights inside that

> **Theme list**: Background is the space with starts and planet. When there is a small explosion during game play, the planets flashing in red colour. When there is a big explosion during game play, whole the screen turn to white colour for a moment. Sound volumes are variable and been louder on small explosions. On big explosion, volumes follow this graph:

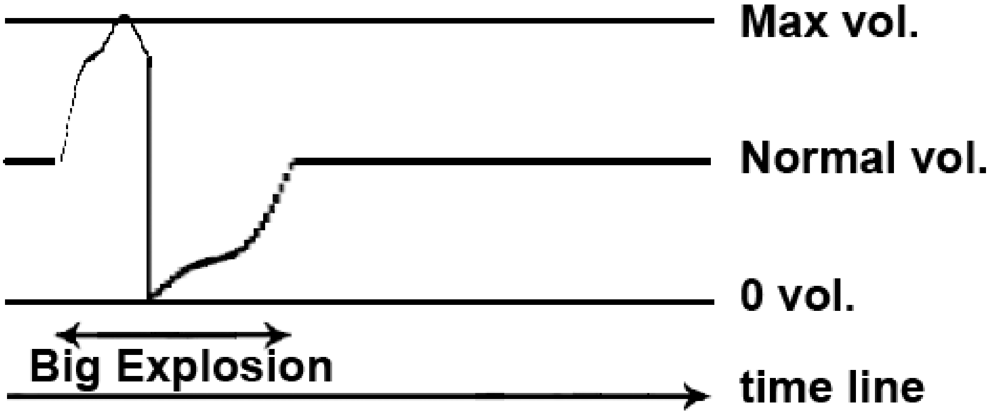


Figure S1-18

2. Items List

On stages and backgrounds of any game, there are some items that player can watch or hear. These items are described by 3 main properties and create a list named "**Items List**". These properties describe the ways that an item changes. Also they describe interactivity of an item in game.

The properties are:

M: Movement of item with all details related to coordination and velocity of an item.

O: Fidelity of an item that presents "Sensitivity" of item in case of other items, player and game stage. This property describes relationship between items in game, how they affect each other, visual opacity and behaviours. "**O**" is not only for graphical items, but also can apply for audio items.

P: Percentage and numbers of an item is presented by this property. For sound items, "**P**" presents frequency and loudness of an audio item on the screen and also number of channels.

Including these list we can mention which item is related to these lists. When there is a game based on true story, these lists are very important.

Example Seven – information unit A - Starting airports

Item List: Pilot's cabin
Necessary controller: Fuel, Height, Radar
M) Fuel: Digital format presentation by 100lit.
Height: Numeric analogue presentation by Ft.
Radar: Full digital presentation by scan speed on 1/2 per second.
O) Fuel: Green for ideal, Yellow for middle and Red for lake
Height: On dangerous height, there is a yellow alarm light.
Radar: Always online even on the airport
P) Positions are like figure S1-19:

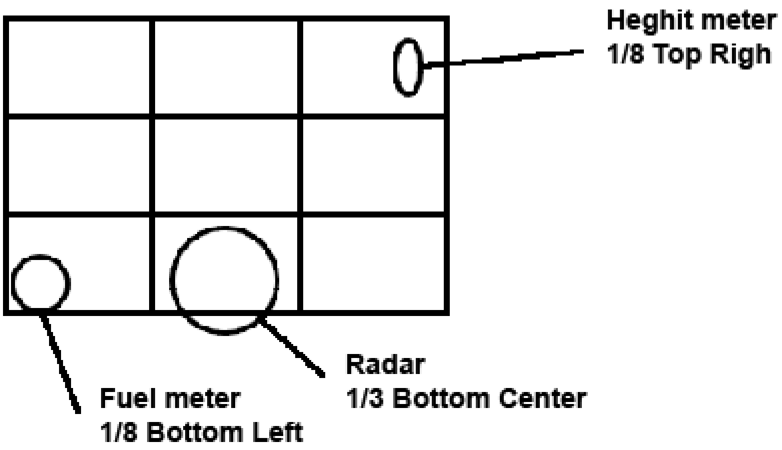


Figure S1-19

Example Five – information unit Z - End of game and a happy little girl

Item List: Happy little girl

Object: Little girl

M) Moving face fast. Big smiles and rapid eyes blink because of happiness. Sound of smile is high quality.

O) Colours on mouth and eyes are bright and the brightness is increasing by passing time. Smile sound start from less loud to loud and then music start. (Manga theme)

P) Positions are like figure S1-20, camera start from scene A, after one rotation around head, reaches to scene B. Zoom changes from 1/9 to 1/3 of whole the stage.

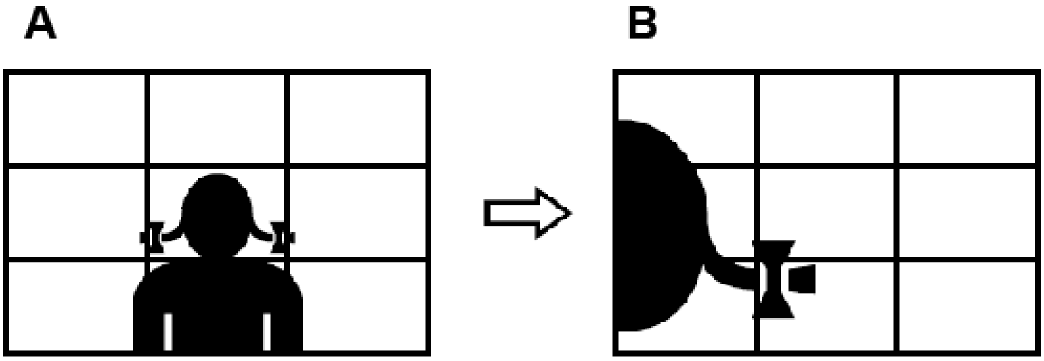


Figure S1-20

3. Environment Methods

Environment methods make disciplines for information unit. By understanding **environment methods** by player, he can get information and understands data from the game. **Environment methods** describe angles of visual items, duration of them, moving lights and random actions. **Environment methods** are very important because they contain and describe the logic of “game play” for players. This list will used by programmers for creating logics for game on next levels during **Nibok** method.

Example Six – information unit A - Small farms with $1000 credits

Environment method list: Global visual angle is 45` for all axis (see figure S1-21).

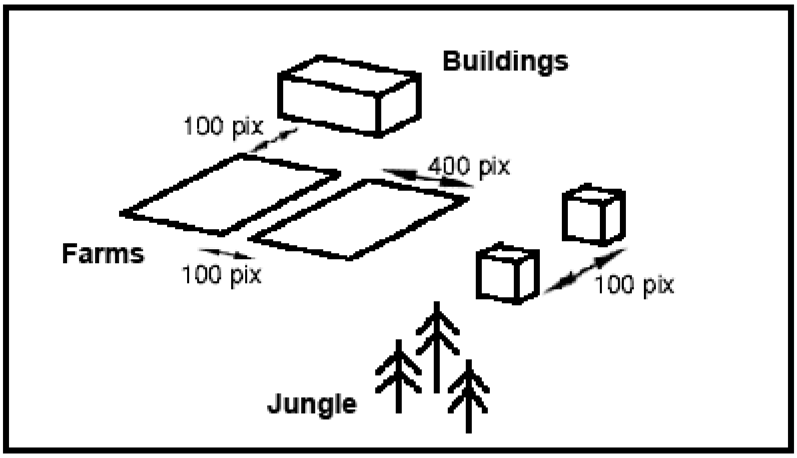


Figure S1-21

Ambient sound is wind and bird's sounds. Minimum distance between items is 5 pixels.

Shadows are visible and short. Winds visually observed on farms. Other moving items by wind like flags have 50% less speed than farm's wind effect. Thus wind speed on trees has 50% less speed than farms.

Example Six – information unit D - Shopping centres A: Local and small

Environment method list: Various colours on items. Passengers are more than wheel –based vessels. Sounds of children are louder in crowd sound s. Shadows of clouds are visible on the shopping centres. The environment is "Warm" and "Busy". Maximum building's heights are 2mt.

Note: "Nationality" and "Architecture" of this location is listed in "**Theme List**". I put "**Theme List**" of this information unit here to show the difference between "**Environment Method list**" and "**Theme list**".

Example Six – information unit D - Shopping centres A: Local and small

Theme list: These shopping centres are like "Bazzar". It has a roman architecture with roman style dress on people. There are god's statues on 4 main directions of Bazaar. They use gold for money system. Some people are running but others have a mass normal speed. There are some chariots in there for transporting products.

More examples:

Example Six – information unit C - Shopping centres B: Far and big

Environment methods list: Colours are in range of white to blue. There are some streets in the environment and there are many wheel-based vessels in there. All items related to ambient sound, have same volume. Clouds move slowly and their shadows are visible on the stage. Building's levels are 4 in the centre of city and 2 in the suburbs. Styles of movements for objects are "serious" and "logical".

Now there are two theme lists for this example. Each of them can be used with environment methods list but they are different:

1- Choice one:
Example Six – information unit C - Shopping centres B: Far and big
Theme list: Whole the stage follows "Greek" style architecture modes. There is a big temple in the centre of each colony (city, shopping centre and ext.). All roads are finished to the temple. There are 4 main roads, first one....

2- Choice two:
Example Six – information unit C - Shopping centres B: Far and big
Theme list: The story happens in China around 500 B.C. There is a temple on the north of the game stage. Depends on how big your colony is, the temple grow and came bigger. There are 2 main roads from city/bazaar to temple. There is a forest on the north....

In fact, **theme list** presents bones of information unit and **environment method list** provides the logic of interactivity in an information unit. First topic that players understand and learn is from theme lists of the game. Theme lists create a live character on the logic that describes by environment methods list. Theme list is important in **Nibok** method by other points of view. They can be changed during game play and change an information unit A to information unit A`. They can be growing logically, linearly, dynamically and genetically. They describe all properties of an object (information unit) and based on that, they can generate, transferring and affects on data from player to the game engine or to other players on a LAN party. On example Seven, the tools on the pilot cabinet are same but changing weather, fog on airport and many other items related to navigation can be set to player expertise and make the game more excited for him. At this point you can compare many games that are same as information unit but are different on themes. A good example is "Total War" game series.

4. The Prodigies

On all information unit there is or there are some special items/effects that prepare for a special purpose. The purpose could be related to beauty of game, more information related to story; upgrade a topic for player and some sort of things like that. The concept of these prodigies and relationship between them to other properties in information unit are separately prepared in a list named "**Prodigy list**". These prodigies are independent, they can use or mix any logic and concept for game. This list can be wisely designed and prepare some sorts of modes for player. They can affect player directly and attract him to playing game more.

Example Six – information unit D - Shopping centres A: Local and small

Prodigy list: On bad weather situation during game play, there are some thunders with high sound. After hearing thunder's sound, all animated element on the stage freeze for 1/3 second. The volume of thunder's sound is using multiple 2 of a logarithmic function. This function is based on research for brain waves...

Example Four– information unit B - Fight area

Prodigy list: If there are 2 big explosion in same time, visual stage change to inverted flash colours for 1.37 seconds.

There is no limitation for concepts on **prodigy lists**, but these points are important for make them more powerful and special.
-On childish games, prodigies are very popular. Children love prodigies during games.
-On serious and bloody games, sound quality is very important at the moment that a prodigy occurred. Sound quality is like a frame that makes a prodigy brilliant.
-On sport game, it is better to preparing a prodigy depends on weather and earth situation and not broke the normal rules of game.
-The player believes that a prodigy in game is prepared by game engine and is noted inside the game story, but, if players can "**Make**" a prodigy during game play, they never forget that moment and that game. This is a wise tool that used on some online tournament first person games. "Shoot and Jump" techniques some time create a prodigy situation by expert players.

Preparing lists for information unit is finished by preparing this list. Now I talk about "**Rule Units**".

Rule Unit

Imagine a player is playing game. He uses mouse or keyboard for making reaction to the situation that happed for him in the game. Mouse and keyboard or game-pad is the tool for sending players information to game engine. This information is the results of what happen in the game for player's items and objects. Game engine receives this information and prepares next changes on game stage. The "**Rule Unit**" is responsible for all of these responds from player to game engine and from game engine to game stage. This is the main responsibility of "**Rule Unit**". Actions like attack, move, save and Load data and select items are defined in this unit.
Another responsibility for "**Rule Unit**" is describing how these reactions must show in game by engine. For example there is difference between shooting and crawling but both are in "Move" class but they are difference. There is difference between shooting a normal rifle and a sniper rifle but both are in "Shoot" class. There are some related topic on designing and creating "**Rule Unit**" that listed here:
- Present and assign inputs for playing the game: List of all main/functional keys for controlling game, all gaming navigation tools, save and load game and reply or capture functionality on game. Special control options like multiplayer controlling extra features.

-Position and Calculation for player: Position of items and objects related to player's action during the game like: Fuel, ammo, food, gun, keys... Calculation of related situations for player's action during the game like: Environment, Enemy and friend status, un-changed situations
- Position and Calculation for others: Position of other objects in game stage related to player's action like: Enemy gun's bullets, damages, changing or morphing items to other items.
-Best Hardware choice: Describe best situation related to gaming tools for players. Synchronization on multiplayer mode and give some reports for choosing best console to develop game on.
-LAN: Preparing reports and logics for multiplayer and other LAN activities by player for game. They can create digital version of some team games. Choosing or design LAN protocols and programs depend on game style and situation as more technical concept.

Rule units use these concepts for develop:
1-Character Concept: Figure and sounds for enemy, friend, independent objects in game and other items related to them.
2-Arrangment: Design lines and methods for moving player through game. Preparing amounts of enemy and friend units. Provide rules for ending game in easy or hard mode.
3-Syncronization: Timing all elements and items in game. When the sounds are fade or play. When player's resources will finish?
4-Network: Protocols and game standards for parallel actions on network.
5-Data: Load and save data during game play. Some hidden data that player can discover.

Rule unit has calculative and actual presentation. It describes the real-time playing of game before the game developed. **Rule units** are like artist who play on a stage named information units. They both create game.

Example Five– rule unit m - Methods of moving to the castle and fighting with enemies

Description: The little boy character can move on the earth surface and between trees. His weapons are archery, fist and magic thunder. There are some arches between trees that can be collected by him. He can use fist for close enemy units. Enemies are in numbers but they are week. The road to the main castle is straight. Inside the castle enemies are changed and their powers are increased. Also he gets an additional weapon named "magic wood" that prepare flying action for him. He can also shoot when he is flying.

Example Six– rule unit m - Farming methods

Description: Farm product is "Grain" that present by number on the top of screen. "Water" and "Fertilizer" are used like resource by player and presents by number on the edge of screen. If player has one farming tool, his product speed is multiple by 2 or 4 depends on what kinds of farming tools he use. It can be simple or and industrial machine. Weather situation affects on production. Products move to factory by some transporting system and

change to gold for player. He can use gold for upgrade; buy new/more farming tools and expand his farms.
Water can be prepared by using well. More wells bring more water for farms. They can be upgrade by using pumps. This upgrade could be between 2 and 8 times more than a normal well.

Rule unit describe game play for game providers. It focused on technique and concept of “Control” for game and gives some roots for designer on the other levels of **Nibok** method for how to develop that game. This unit has something like a multimedia and interactive functionality between player or players related to game and game engine and this is important. How many players can join to the game and numbers of logics that can apply to game engine, are the roots that create in **rule units**

Analyse of rule unit

For preparing details related to a rule unit, we need lists including description of rule unit. These lists contain all details of each rule unit. There are 4 lists for rule units that MUST be prepared one after another. This guarantied clearance and performance of details and save time in big projects.

These lists are:

1. **Characters list**
- Friend Characters description
- Enemy characters description
- Other characters description

2. **Variable list**
- Numbers of variables in game
- Timing variables and their properties

3. **Game’s logic list**
- Move and company rules
- Death and rescue rules
- Special situation rules

4. **Network list**
- Describe Data sizes and standards
- Describe Send and receive protocols

Note: These entire lists will be used by development team and they programme and design game based on details in these lists. More detailed and clear lists improve performance and quality of development and prepare better game design on next levels in **Nibok** method.

Here is detail for 4 main lists of rule units:

1. **Character list**

   Anything that effect on player's object or player's objects is named as characters on game's stage. These characters are friend or enemy or independent objects that affects on player situation during game play. A character designer must be aware of big ranges of artistic, psychologic and social topics for design/describe a character. Design concept is free for designers but some features are related to this topic:

   -Emotion: Rudeness, kindness, mysterious, hero, evil, normal...
   -State: Move style, visibility or invisibility, sounds, healing or have problem...
   -Connection style: Character's connection to game stage, character connection's to player.
   -Changes: Evaluation and life cycle of a character during game, sizes, speed and tranquillity...

Example Five– rule unit m - Methods of moving to the castle and fighting with enemies

Rule unit m – Character list:
Name: Tom
Description: This is main character of game that navigated by player. He moves fast and when he jumps, he reaches to half of main stage's height. His colours are special and fast detected in the screen. His face is a bit angry during shooting or fighting. He has no change during the game. If his energy changes to low, his clothes seem damaged. If his energy finished, he faded from stage and appear on the edges again.
Name: White Dragon - Tasu
Description: This is an enemy character. Its size is 2 times bigger than Tom. It shoots fire in close distance to Tom. Its sound has same volume as other enemy units. It moves slowly and can jump over the items. If Tom destroys it, it flashing on the screen and then disappeared.
Name: Stone
Description: this is an independent character and has no effect on other characters. They are in 2 main formats: simple and planted. Simple stones have simple design and other animated objects (like Tom and enemies) can jump over them. Planted stone are unstable during move and change the speed of movement randomly. Planted stones affect sound of objects which are passing over that. ..

Character list has some concepts for graphic designers and sound providers. They use this list to create real characters and real audio files on next level which I will talk about that on next chapter.

Example Six– rule unit q - Sending products to shopping centre by normal transporting system

Rule unit q – Character list:
Name: Transporting animal
Description: These characters used for moving grains to factory. They are "horse" or "Camel". The player's colour or sign are presented on them graphically. If they were treated by bandits or wild animal they move 4 times

faster and if they were killed, they fall on the earth and disappearing during long time. The grain that they hold will already lost after they were killed.
Name: Bandits
Description: They move on group of 3. Their moving style is like move-stop and again move-stop methods. If they attack to something, they generate some unusual noises. If they damaged, their speed will be low and if they killed, fall on the earth and disappearing during game play.
Name: Grain transport vehicle type A
Description: It is a simple truck with tent on back. Tent has colour or logo of player. If it damaged, tent has some fire and if it destroyed, there is a simple explosion and it separate to some small objects and disappearing during game. It can hold a specific amount of grain and if player upgrade it to type B, they can hold 2 times more grain but their speed became lower.
Name: Tree
Description: they are in ranges of “Pine”, Willow” and “Cypress”. When winds blow, they have small movements. When there is fire accident, they burn and then morph to a black burnt wood and remain until the end of game.

After preparing character list, we can prepare variable list.

2. Variable list

In this list there are numbers, amounts, prices and other numerical issues related to characters in game. Materials, weapons ammo, magic’s amounts and some sort of things like them are described in details in this list.

Example Five– rule unit m - Methods of moving to the castle and fighting with enemies

Rule unit m – Variable list:
Name: Tom
Variables list: There is only one character of Tom in the stage. His energy is 100 units. If his energy reaches to 20 or less, there is an alarm system for present the situation for player.
Name: Black crows
Variable list: They are “always” on the stage with numbers of 2 -10. They can be destroying by one shoot. If they reach to Tom, they will disappeared and decrease 1 point from Tom’s energy.
Name: Golden Stars
Variable list: They are in groups of 3, 5, 8 and 10.They appearing by 1, 2, 4, and 8 in time scale of game (every 1 second or 2 second or 4...). Total numbers of them are 100. Only Tom can receive them. They destroyed by thunder and from every 3 starts only one remains.
Example Six– rule unit q - Sending products to shopping centre by normal transporting system
Rule unit q – Variable list:
Name: Transporting animal
Variable list: Their energy is 100 points. For each two shots from enemy unit they lose 1 energy point. If they reach to 4 points on energy, they escape from that position to a random place. Player must control them again. If they

do not receive any damage, their energy increase by 1 points every 1 minute. If they receive damage from stones, they lose 3 points energy.
Name: Caravan protector
Variable list: They have 20 points energy. For each three shots from enemy unit they lose 1 energy point. They can be healed by medical tools and reach to full energy again. In raining situations in game, for each two shots from enemy unit they lose 1 energy point. If they have an armour upgrade, for each five shots from enemy unit they lose 1 energy point.
Name: Tree
Variable list: their energy is 50 points. And when they reach to 20 points they burn and lose energy until it reaches to 0 point in 4 minutes. It can damage nearest objects with amounts one points per 10 second on energy.
Name: Cyclone
Variable list: Cyclone has 100 energy points and can destroy objects in time …

3. Game's logic list

Patterns of movement, fight, increase or decrease fees, special movements like reloading a weapon, simplicity or hardness of game, AI methods and other algorithmic topics. This list is an important reference for programming the game logics, for more benefits on development we can use vast series of physical and mathematical formulas and functions.

Example Five– rule unit m - Methods of moving to the castle and fighting with enemies

Rule unit m – Logic list:
Name: Golden Stars
Logic list: Their positions are fixed. They rotate by period of 1/36 per second.
Name: Black crows
Logic list: Their move-paths follow ellipse formula. They move to Tom position from top to dawn. Their speed follows power two formula (y = x^2) and they lose speed after appearing on stage. If they are not targeted by Tom, after two minutes they rush to Tom directly and their speed will multiply by 2.
Name: Big Trees
Logic list: Their position never changes. If “White Dragon” appears on stage, they move 5 pixels to left and right slowly. The effect of this slow movement cause falling all objects which are locating on them.
Name: Tom
Logic list: ...

These lists must prepare one after another, first one is Character list and second one is Variable list. Then Logic list is third list that cover Rule units more and finally the last one, Network list covers whole the topic of networking for game developers and players. This discipline for creating 4 lists caused an automatic arrangement of information related to Rule units for lists. Developers can follow the features more clear if you prepare lists in that way.

Example Seven – Rule unit r - Special Weather's method

Rule unit – Logic list
Name: Type I – Clouds
Logic list: These clouds are posed in the centre and bottom of view. They have fixed size at first and then, when airplane move to them they change bigger, until covering whole the view. This change in size following formula: H = n x t ^2 H = size, n = large scale, + for moving forward and – for moving away, t = time
This a simple object of cloud and can provide more complex clouds with that, in addition, t can be affected by difficulty of game or experiences of pilot(player).
Name: Parasites on sound of airport control tower
Logic list: This is a noise character and can changes between 1 KHz to 12 KHz and play over the main sound of airport control tower. It's volume is depends on weather and damages on radio unit in airplane. The formula for...

Logic list can use the reasons of a specific research. On atomic explosion example which mentioned before, all rule units can follows real formulas that prepared by researches. Sport games there are lots of real formula which used by developers and make game more realistic and beautiful. Economy games use banking and business formula which provides good attraction for pro-gamers who like real economy based games.

All these tree lists are prepared for each character and in some cases describe additional characters (like sounds and changes on situations). Now we can prepare the final list.

4. Network list

Many of new web-based games are multiplayer. Most of them do not have a complex game engine but are powerful on network and multiplayer modes. Because of that, there is Network list related to Rule units in **Nibok** method. If multiplayer mode is essential topic on game, all methods and standards related to multiplayer mode and other networking issues must be listed here. This list contains general and specific data for each Rule unit. It covers actions like finding, fighting , join and healing other players by player; multiplayer rules like death match and capture the flag; massaging during playing game and other network issues that creator of game story describes for developers.

Example Four – rule unit r - Methods of moving space ship after dropping bomb

Rule unit – Network list:
Name: Main spaceship
Network list: Player's spaceship damages when impacts with other player's spaceship. The amounts of damages are equal between ships. In easy mode, friendly fire does not affect players, but in hard mode it does. Maximum players are 6 in stage.
Name: Meteor

Network list: Only one platform in network analyse the paths and movements of these meteors. Ping is not important for displaying them because they have not any damage or benefits for players.
Name: Yellow spaceships
Network list: These objects are important and must be synchronize in multiplayer mode perfectly. They must be targeted by players and move with fewer pings in network.

Conclusion

Previous chapter described first step of **Nibok** method. In this step, the story of game is described by a game logical graph. Game logical graph is something like flowchart but it is interactive. It can shows stages of the story and actions of players in the story. By defining some lists, the first step of **Nibok** method is provide detailed information related to game story which will use for next steps. The first step of **Nibok** method is known as “**D level**”.

Chapter Two – Definition steps - C level

Introduction

There are detailed lists that contain lots of data but related to one topic, “The game Story”. This level follow steps throw make all this details, alive. People who are expert on graphic design, sound and music, history, math and many more can join to this level as a part of game development project.

All lists came from last level which known as “**D level**” and described in chapter one. Now in chapter two, game development move to a new level known as “**C level**”.

Overview

Before talking about how the lists changed to a developed game, let's see how an overlook to **Nibok** method. The **Nibok** method not also can provides detailed information for designing game, but it presents a useful business model for designing game. This figure shows how many levels are in **Nibok** method:

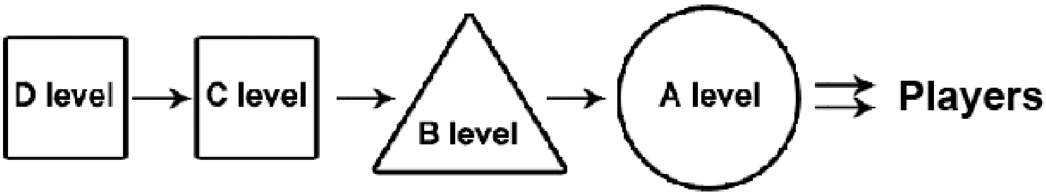


Figure S2-01

Chapter one covered "**D level**". In "**D level**" the story of game is created and based on that, all details related to game is prepared as lists. Everybody can join to that process in "**D level**". After finishing "**D level**" by preparing all detailed lists, developers can presents lists and other detailed information for getting credits from client to move to "**C level**". In "**C level**" they (based on lists) create images and sounds and animations for game. Then by gathering credits from client they can move to "**B level**" which is more technical. Here in "**B level**" the game is programmed by programmers. After final meeting with clients and gathering final payment, developers move to "**A level**" which is testing & final editing and business issues. Then game sends to market for players.

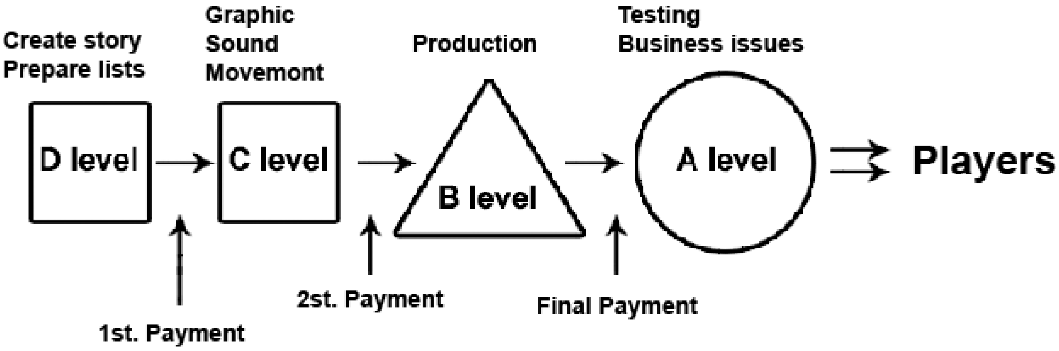


Figure S2-02

As the figure S2-2 shows, there are 3 milestones for payment in project. First time, client(s) accepts the story and pay fees for starting the project. This is good because everybody can prepare story with detailed lists as mentioned in last chapter and described as "**D level**" for free. The game story can be prepared even by people who have not any experiences in digital world; it can cover any story and also can be adapted to any real story. "**D level**" is a free start for all.

In "**C level**", there is one or there are some groups who are expert on modelling, graphic design, narration and sound. They prepare whole the material that will used for programming game. These materials are EXACTLY based on lists from "**D level**". At the end of this level, whole the materials presented to client(s) and after acceptance, gather second payments for following development on "**B level**". This chapter describe "**C level**" whit all details.

In "**B level**", game is programming and programming team used all materials from last level. They make the game's details in this level with all details and data related to the story. At the end of this level, client(s) will present by the beta version of game. Then client(s) pay final payments for game developers to move to next level which known as "**A level**".

In "**A level**", game is tested and preparing some advertisement before lunch to the market. Other global business issues related to game is contracted in "**A level**", because the game business is depends on many real-time features that is not in the scope of this note.

All these levels could be done by one or by number of groups and because of separated areas of work; **Nibok** method can cover much expertise during making game's details period. This makes finished game better as quality and reach for other related items for game. And, because of separated levels, developers can provide more than one game from same story. For example the story is related to a real story from World War II, and developers can create 2 different game type of that story, one strategy and one first-person shooter game easy. This is like film making. There are 2 scenarios for version of a particular story that directors make them different and because of that, there are different fans and opposition for those films.

"**D level**" is a free gate for all people with any level of knowledge. It can be done by fewer credits. Then there is "**C level**" which is related to people who has more experiences in visual/audio/text items. This is another section for putting credits on project. After that, game will create in "**B level**" and after finishing product period; there is another opportunity for putting more credits on advertisement and marketing of game before it lunches which is one of the highest expensive industry in the world.

All members who work on game development during these levels can be from one company. But for more economical benefits, a game project can be shared between more than one company and they can work on multiple projects at same time. **Nibok** method supports multiple business plan and expertise due to developing game.

Now with this overview of whole the story, I can follow the **Nibok** method into "**C level**".

C level

Whole the lists which are prepared in "**D level**" come to this level. They are analysed in detail and based on them graphical and audio items will create. Animations and other related modelling concept like capture motion data is prepared in this level. Languages which used in game, fonts, interface and mathematical formula are prepared in this level both as digital and analogue items.

"C level" is separated to 4 main sections:

- Sound and graphic
- Data

- Mathematic and statistics
- Game proposals and Director list

1. Sound and graphic

Here with using of all lists from last level, all graphics and sounds are generated. Characters, weapons, backgrounds and anything that is graphical will be designed. For example if there is a character in the story, graphic designers must prepared this character in all needed forms like run, attack, sit based on lists from "**D level**". These designs could be used in games posters on next levels for advertisement and more development.

Sounds of characters, items, backgrounds and special items are recorded or played based on data from lists. They can provide by person or a music band. They can be based on some historical moments or they can be digitized from a real subject.

All developers who work in this section are expert. They are powerful to create real materials based on text ordered based details, so these guide many artists to work on game projects even if they have simple background of gaming or programming parts. By the power of computer hard wares, everything can be possible to create inside game, so there is no limitation for reaching to high quality and best performance for a game.

There are some important points for designing these products:

- Use the power of digital design and freehand design

- Creativity

-Full awareness of detailed lists

-Focused on quality

The materials which provided in this level are some sort of graphics and sound files. They must named by full details and sort in one or more than one tables. For example "Graphic-Tom – attack mode – farm 01 – size 100x74 – Photoshop file" or "Sound – Red Dragon – attack mode – take 02 - 44.100 kHz - wave file". The naming structure and methods of these files is depends on project's modes and style and there is different between large projects and small projects. Some group prepare their material based on apple computers standard and some prepare based on Microsoft operating system, other work for PS3. In large projects, companies have their own standard for this purpose. All of work is based on detailed lists and they can use each of them. "T" that represented by I + R = T formula can help to estimate how project is big.

2. Data

In many games there are some data related to characters, game situations, extra information, date, special massage... that will use by player in game. These data are related to many topics like economy, history, story of game and so on. These data must be research and provide based on lists. Developers can use any resources related to topic for providing these texts or documents. These data are separated in some main categorise:

1-**Help for game**: Every game has a help section. In this section players know how to play the game and what the goals are. Here in "**B level**", developers provide first series of "How to play?" help section for players. This section will complete in next level by programmers.

2-**General information**: All additional information related to game and items inside the game is gathering here. They cover items on the lists and create an information section for including in game. Some games are based on historical or scientific aspects. In these games, players can learn new things from the game. The game is like a book that player enjoy to be interactive with that (play that) and learn new things. Some games are not using large amounts of data but steel need some documentation related to the story for players. Gathering these data based on lists are important because it cover the action of characters in game and describe the functionality of each item in game.

On the other hand, these data can be used by other members to make game's quality higher, especially when the story is based on true story. Also fans like this data and use them for many purposes. Hero names are some example for this section.

3-**Puzzles and answers**: Some games need some puzzles and answers. Players follow some puzzles in game and must find the answer based on that information. This can be done by peoples who are expert on designing puzzles based on texts and icons.

4- **Science fiction data**: Many games use unreal stories in an unreal world. Some data like magic, imaginary concepts, Wizards, Ancient Empire.. are unreal. For providing these data, developers can mix many sources or create new unreal based text related to story here. This area is absolutely creative.

5-**Real data**: Sometimes the game is based on real technologies in real world. So all data must be real and based on reality. For example in a military simulation game, all simulated data are based on real situation in real world, so the texts in game must be cover this reality perfectly. This guaranties the quality of game. In sport games, data related to players and leagues must be real and well provided by developers because players love these data and enjoy them. The main point in these sorts of data is preparing real "Names". Names are keys to make data reliable and make it as a reference for player's knowledge because many players used real game data as some facts in their life. On some economy based game, there are real kinds of business data that provide game for players. Flight simulation is another example which reality of data can be pointed as source in some cases.

All data must be prepared as computer files with specific format like PDF or TXT. They must be clean and without any errors. If there are multiple languages for game, all the text and transferred version must provided here.

After finish this stage, there is some extra information which must add to the text files:

-**Definition mark**: This is a number between 1- 100, it presents how that file is important in that topic. For example if topic of text files is related to “Astronomy”, the date related to Earth has more important points that other data files in this topic. So it can have higher mark between these files.

-**Capital letters:** Use capital letters for in game’s title everywhere in text file.

-**Font style and sizes:** Describe which font size is better for that information. And include the font shape.

-**Copyright:** Prepare copyright issues for protected texts.

-**Resources:** List of references for texts.

-**Age**: Who can use this data and who can not? Sometime the data is related to a horror topic which is not advised for children.

-**Upgrade**: In some games, data can be modified by users. Like sport games which upgrade data bases related to real world every season. Players like this opportunity in game and this must be mentioned in text files in details. Another example is multiplayer tournaments on network. Some data can be changed during tournaments for players.

3. Mathematic and statistics

Game is a big changeable environment which player changes it when plays that. All these changes provides by logic and algorithms. Mathematical formula and functions are the keys for change game’s situations. This formula activated by CPUs and ALUs on computer hardware and presented by graphic card and sound card. Depends on how these items are powerful, game can support more detailed formula and functions.

Based on this, all formula and functions must have clean results. It means they must be easy to use and fast in generating results. These results are based on lists and the character behaves in the game stage. Here there are some examples of how mathematical formulas are needed for game:

-Rocket’s path follows arc patterns that can be provides by $y = \sin x$.

- Cloud’s move speed is sometimes fast and sometimes slower. This can obtains by random functions.

-Liquids like water waves different when there is an explosion on them. Some game engines support this effect.

-Impact a car with hollow barrels because of enemy treat. This need capture motion for establish best formula for that.

-Economy formulas for business production.

-Wizard magic light and dynamic shadows in stage.

The main point of all the formulas is they must give clean results fast as possible. If there are some errors on result, it must be mentioned and covered in text file. Also formulas must covered difficulty of game in easy, medium and hard mode. This can be earned by definition ranges of numbers which passed to formulas and create output results. Output can be simple text file or code or even graph. Programmers use different languages for simulation these formula and results. Simple and independent mathematical formula can define by any language. Even they can change to some library files for game engine. So preparing a complete list of result, define errors on results and other related topics prepare better performance on programming level. Here are more examples of formula:

-Percentage of mixed materials for preparing new material with them.

-Random selection between cards.

-Find dice number depends on difficulty of game.

-Percentage of mine materials in a mountain.

-Create new entities by using genetic algorithms.

-Percentage of how people are patient on increasing taxes during game play depends on their nationality.

Genetic algorithms are the most beautiful part of this story. Using logical functions related to genetic algorithms provide speed on calculation and naturally they support AI of game. Game engine can earn experiences during game play and prepare more wise and real behaviour for characters. People who are working on these formulas must have good knowledge of fractal formulas, differential formulas and group theories in mathematic. Main point is “wise complexity” of functions not “poor simplicity”. Using simple formula for timing is easiest way in game but using fractal functions for this purpose is hard but has many benefits. It can support all other mathematical processes in game at same time if necessary.

Also define good logical functions can decrease lines of program in game and then prepare better performance during game play. Using graphs for simulation trees and details are useful but the best way is using one unique formula that can prepare “**All**” results that we need for game in each moments of time.

4. Game proposals, director list

After preparing all graphs, sounds and logical function for game, there is another section which is like bridge between this level and “**B level”**. This is most important text in **Nibok** method. In this section, developers define game style.

Using experienced people related to game design is essential in this section. Here after preparing all data and texts, developers must define game style which is not only related to data and texts that they have, but also is related to client(s), business view, era and creativity.

Computer games have 7 main styles. Some games follow only one style. Some use more than one style that is hard to programming and define. These styles are based on player’s needs and technical issues related to digital concept of gaming. Nowadays, some hard ware providers adapt their products to cover issues related to game styles. For example on GPU of some graphic cards, there is some hardware for calculating physical formula faster. This shows how game styles are important for players and also for developers. Here I talk about each game style:

a. First person

Players visit and play game from their own eye's angle. It is like moving in a place with camera on eyes. Everything will move when player is move. He is like a traveller inside a virtual reality. This style is introduced in early 90`s and today have many lovers. The action and reaction in this style of game is individual and each player has his own style for playing.

Games like "Doom", "Half Life", "Quake" and "Ghost Recon" are good examples of this game style.

b. Third person

Players play with animated digital puppets that present their character in game. They can watch their hero during playing game, see how it damage or how it earn power, see how it fights or how it move through game levels. Many children like this game style. Camera angles and changing view port dynamically is essential point in this style,

Games like Sega Marine, Metal Gears, Diablo and many Play Station console games follow this style.

c. Strategy

Players are responsible for creating and commanding colony of items. They must use materials, building new units and factories, assign power and reach to the goals by conquering enemy or obtaining specific purposes. Number of items and materials in game is important. The game is base on how players are wise and fast. There is no limitation on how players choose their own strategy for game. When players play on multiplayer mode in network, they share their intelligence for take over other players or teams. This is the real joy of strategy style which presents players intellectuality.

Games like "Civilization", "Age of Empires", Command and Conquer", "Settlers" and "War Craft" are some famous pure strategy games. Strategy games, on the other hand, are good for preparing vast range of information for players. Economy, history, war and cultural data topics can be easy provide and educate during a strategy game on players.

d. Simulation

At first time, this game style creates for military purposes. Teaching pilots for navigation fighters and bombers during day and night with different weather situation, was the story of first series of simulation games. After digital revolution and preparing home computers, developers prepare simulation style game for driving cars, commanding submarines and destroyers, navigating tanks or even space shuttle and many other topics. "Flight Simulation" is one of the famous air simulation games which have many hardcore players. The main concept for these game styles is, "Show a reality, digitally playable".

Games like “Silent Hunter”, “Destroyer Command” and “Test Drive” are other bright examples in this style.

e. Sport

Sport games are popular. This game style is simply simulating a specific sport on computer with using all real features and data related to that sport. Sport games cover long range of real games like chess, ball games like basketball, driving games like race and martial arts. Sport games follow/use these sections:

A-Team games: Player can control a team of virtual players. He can play with each of his team member and sometimes he can make them experienced in a particular techniques related to that sport. “Fifa”, “NHL”, “Speed Ball” are some examples for team games.

B-Rally: Player drives a special car. Depends on game type, he has a rally match, racing match or special type of match during game play. He can choose or tune his car, or buy new one. ”F1”, “Need for Speed” are some examples of these games.

C-Personal experience: Player is the only one who plays the game. Chess, card games like Poker, martial arts, ski, snooker and other sports like them, are played by one person in real world. Games like “Chess Master” and “Mortal Combat” are good examples in this case.

D-Coach and Management: Some sport players like to be like a coach or manager of a team or a group of players. They like to educate their virtual players, make some friendly match for their team and finally move to serious tournaments and earn more points. This is how the games like “NBA Manager” are famous as coach and management game.

f. Adventure

Adventure games are one of the oldest versions of game styles. Players must solve different kinds of puzzles which are visual or based on texts or even based on some sound data. They are like Alice in a mysterious virtual world which must find the way to the game goals. Designing these games is one of the purest topics in digital game design. Many films use this style to provide game version of the main scenario.

Games like”Monkey Island”, “Myth” and “Beneath the Steel Sky” are some example of adventure game style.

g. Freaky Game Design (F.G.D)

Games have their own style that cannot be found on other games or use mixture of other styles as a unique style. I cannot describe what specific concept is used in this style but I can say, this style is memorable more than other styles. Players never forget how they play game and how was that. “Tetris” is a good example of F.G.D game style. It doesn’t follow other styles but also has its unique style in game design.

Other examples: ”REZ” on PS2, “Worms” and “Arkanoid”.

“**Director List**” or “**DL**” is document that include to previous three documents. This Document suggests game style and prepares business plan related to players of game, areas, network and other related business eras. Also it describes game during playing time, in fully detailed information. Detailed information covers technical issues of cinematography, sound, game concept, demos, psychologic items and art issues. All related data to game like extra movies, sounds and additional data requirements (like copyright data) which must obtained for game, are mentioned in detail in **DL**. This list defines game for production. DL is core document of **Nibok** method.

The **DL** document follows these sections:

-Define philosophy of game.

-Detailed information of suggested game style/styles.

-Describe game play as players view (both normal player and expert player).

-A list of all items that prepared in “**C level**”.(current level)

-Define and explain suggested hardware and consoles for game. Provide additional details for special hardware or console for advanced game play.

-Define difficulty levels and logic for game both technical and psychological.

-Parental lock on game and related issues.

-Network gaming details.

-Special suggestions and important points as article or researches.

-Has this game another version? How?

There are different ways for preparing **DL** from lists on “**C level**” and there are more different ways for preparing those lists from “**D level**” data. It’s like a tree of information, which start from leafs and slowly reach to the root as one unique concept. This concept is “Game” that is connected to all parts of that tree of information. More leafs is equal to more details and data. More details and date is equal to higher value for “**T**”. And higher value for T means more time or energy for game.

Now there are 4 sources of data which are ready to move to next level. These data are detailed designed, written and provided based on game story. In comparison to film production, game requires more data and resources. **DL** is similar to scenario in film production but contains more detailed situation and action.

Conclusion

This level of **Nibok** method includes more experienced features to game design. Art and Mathematics, Sound and Music, History and Concept are some examples of topics which are cab be used or experienced here. On the end of the level, all information gathered and functionalized by game proposal which known as **DL** or **Director List**. This is the core list of **Nibok** method. This document is a reference for create game on next level. It can be saved and used for next developments. It is like a scenario that can be developed by different business icons.

Chapter Three - Production steps - B level

Introduction

The computer game will produce in this level. After preparing lists and related data, here we can create game with a team of programmers and experts on game product. Maybe you think previous levels are long and it is not necessary to prepare all these details for a game, but the main point is, game industry is one of the top businesses in the world. Because of that, good business in this area needs high quality items and sufficient discipline for creating game. **Nibok** method, automatically support high quality and step by step detailed information. This guarantied reach database and background for a game. Using all possible expertise, detailed and creative information and strong points of view prepared for game, make the final product not only suitable for players, but also good business plane for developing in future. These subjects are clearer in "**B level**". Here man builds the thing that he dreams of.

B level

All lists which prepared on “**C level**” and **Director List** (**DL**) will use for product process in this level. Before explain this level, I want to prepare an overview of this level.

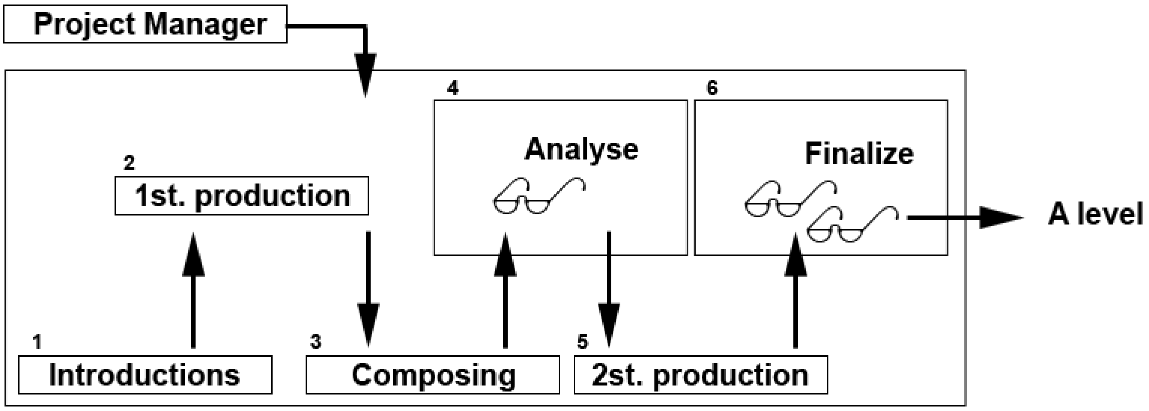


Figure S3-01

This level has some steps that game is products during these steps.

1. Introductions

First step for producing game in this level is hiring one **Project Manager**. Project manager has experience, knowledge and individual points of view over sound, digital graphic, concepts of programming and game. He must has play experiences over many computer games, aware of philosophy of those games and be expert in case of playing games. He is responsible on further stages in this level. He is the master of this level and all teams must obey and follow his orders on producing game.

Second step is hiring **Supply Manager**. Supply manager is responsible for hard ware, soft ware, work place and all items that will used by teams during producing game. Also he is responsible for repairing computers, physical network items and resources which will use for production game. He must be experienced on hardware engineering and other related topics to software and operating systems.

Third step is hiring **Network Manager**. Network manager is responsible for all sections and topics related to internet and networks during production period. He must be experienced on web development and other related software topics to web. He also is responsible for preparing net-advertisement before lunching final product. This is essential for computer games like other advertisement s in other business areas.

Here is summary of these personal:

Project Manager	Managing the project
Supply Manager	Preparing hardware and software for team and also repairing topics
Network Manager	Manage and prepare network programs and related topic to internet

Final step is hiring team/person that has expertises on these eras:

-System programming, professional programming

-Digital graphic design which cover 2D and 3D graphic and capture motion

-Sound engineering and music composing whit experience on digital media

-Network programming.

-Special effect expert whit experiences in film, photography and artistic interactive media

-People with experiences in digital game concept and topics related to game story.

After making contract with these teams or people, **5 main teams** must establish:

- Programming team
- Graphic team
- Sound team
- Network team
- Play ability team

All these section MUST be completed step by step before starting production. This guarantied less time and fees over production period and also increases focusing on quality of final product.

2. 1st production

The list that prepared on "**C level**" is used for programming, sound and graphic teams. They work on these data, product new items separately and save all production as digital files.

- Graphic team create graphical items, they could be 2D, and 3D and motion capture data.
- Sound team prepare music and sounds related to data on list.
- Programming team create/choose game engine. The game engine can prepare game environment that is detailed on Director List. They make game without graphical items and sounds. They must include all formula and logical concept of game here. Preparing memory usage and operating systems issue must be solved in this level.

All teams **MUST WORK SEPARATELY** in this level. Soon I explain the reason for that. Project manager monitors all group`s work progress. He gives solution for problems to each group and support clearance of orders in **DL** and other lists. Other groups have no responsibility in this section. After this level there are 3 main series of files that prepared by teams, graphic files, sound files and game engine files. Only project manager know all details related to these files. All files follow unique concept that detailed on "**C level**" lists.

3. Composing

Here all files are combining together. Graphic and sound teams send their files to programming team and the program team execute and use them in game engine. If there are some technical problems, graphic and sound team must remove the problem and create new files again. Project manager is the key of communication between groups here. His duties are heavy and he is responsible for performance (not quality) of game in this section.

The product of this step is first executable version of game. This is first sketch. Then network team working on multiplayer and network issues and fix it. In fact programming team create skeleton of game in last step and then here, it going to be alive with sounds, graphics and network. Removing problems related to timing between sounds and graphics, network issues, memory allocation for files and speed performance is other functions that must be done in this section. Project manager's experiences on computer games play essential roles on preparing new files, upgrade old files and other related topics of programming.

4. Analyse

The game controls and analyses (not test) by play ability team in this section. Their duties are playing/watching the game and prepare lists of new ideas related to each section of game. All sounds, graphics, special effects, videos, external links and data, game concept, network issues, new items or upgrade items that need for playing and other issues monitor by this team in this step. They also can use "**D level**" and "**C level**" lists here. The team members are not only experienced player but also other related expertise to game like military personals, economist, philosophers, artist and film directors. They can be same persons who prepare game story on "**D level**". Using a creative team in this section is highly recommended. This can guarantied the quality of game in future. Project manager is responsible to list "**ALL**" creative/technical issues who prepared by this team. These issues covers all game aspects like sound, graphic, story, play ability, data and texts, special effects.

This section is like "User experienced on Quality" of game and is important for business issues related to game in future. Creativity is the key on this section.

5. 2nd production

The game will upgrade to new version by applying new issues that prepared within last step. Project manager is close to the programming team on making this new version. Other additional data and items like help files, must include to the game within a specific time table. Now game is growing to final product and forms its special and unique character. There is no testing period on games functionality in this level yet.

6. Finalize

The game is controlling and analysing (not testing) by all groups, companies who order game, advertising companies and other related business companies to the game. The game is analysing for economy planes, cultural aspects and "Best time for lunch" for players. Prepare reports on magazines and lunching game website are some development plant that can provided here. These are some issues that are related to this step:

- Game attraction for players
- Age of players and related topics to that
- Parental issues related to violence in game
- Business plan as international or local business of game sales
- Psychological, scientific and narrative aspects of game
- Advertisement strategy for game before lunch, web site of game
- Contracts and copyright

- Fix problems, final production (problems like copyright, names and so on)

Now all issues related to game been finalized and production period has been finished by production teams. Some screen shots and other advertisement materials is prepared for game and sent to related companies and fans. All standards (like parental lock) been applied on game and it is ready for final level which known as “**A level**”. (Take care for robbers!)

Related issues on “B level”:

- There is no limitation on rules over team’s work during production and they can prepare game requirements based on lists with any methods.
- There is an automatic progress on production and game is more completed during steps of work.
- Quality of game is high because of including all details.
- Advertising for game is on time, so players are not waiting for game long time.
- Programming on game is continual and strongly useful.
- Core management on project by define project manager
- All members are well organized and supported. This makes performance accurate.
- Other aspects from expert people (like artist, economist ...) are included in game sequentially and this is one of the most benefits of **Nibok** method.
- Projects can be shared as international project. Clearance of duties in this case is essential and project manager is responsible for that.
- Game story is the same as prepared in “**D level**”, but there are more details that support it in many cases like cinematography, game design technology, playing experiences and...
- Wasting time and credits are small.

Technical issues:

Programming team:

- Knowledge of operating systems.

- Expertise on programming languages.

- Anticipation on hardware issues related to next generations on hardwares like motherboards and graphic cards..

Game engine:

- Load and save game data for players.

- Optimal and useful mathematical formula and logics.

- Security issues for playing game.

Graphic team

- Technical usage and expertise on using 2D and 3D softwares.

- Using capture motion

- Knowledge on black & white and colour image and their effects on players.

- Optimizing methods for graphic files.

- Level design and good knowledge of architecture and interior design.

Sound team

- Technical usage and expertise on sound softwares.

- Knowledge on effects which arise from mixed frequencies of sounds like 3D sound effects.

- Volume and issues related to effective sound volume.

- Psychology of sounds and music for game as game music, unit sounds and item's sounds. (Very important)

Network team

- Technical knowledge on multiplayer game concept.

- Error handling on network.

- Synchronization techniques and logics on network playing.

- Creative methods on using network protocols.

Play ability team

- Wide and open mind judgment on game and game features.

- Use professional experiences related to media.

- Use professional experiences related to game story and concept of story.

- Have experiences for game play with over 50 games.

- Creativity without attention on hardware limitation. (Because there is no limitation for game design)

- Knowledge on game industry business and other companies.

- Background on special effects.

Other points:

- Choosing media for released game. (CD, DVD or online web site)

- Installation issues

- Patches and upgrades.

- Preparing some suggestion and introduction on next version of game by project manager.

- Define how players can change and personalize game. (Like change graphic resolution and individual setting).

- Prepare screen shot of game for players

- Prepare downloadable demo versions for game.

Conclusion

The final product of this level of **Nibok** method is final version of game without testing process. Now all lists and detailed files create a game based on the story. This is obtained by teams and groups who work together and managed by project manager. "**B level**" provides step by step development for game based on experiences of teams and data from last level.

Chapter Four - Test step - A level

Introduction

Each computer program may have some technical problem that is hidden from its developers. These problems sometimes arise from different operating systems and sometimes occurred by different hardware that used by players. Game is a program that uses operating system and hardware in special way and is executing by players many times. Having a test period is necessary before releasing final version of game to market.

A level

The produced game is testing in this level. For this reason, some pro-gamers test game on a specifics time table. They play game, finish it, and do any kinds of checks for all possible items, options, situations and mode of game more than one time. Game logic, fluency of playing and story line is checked by testers on different operating system by using different hard wares. They are monitored continually by project manager and the problems are reported to programming team by him. If they found a big problem that needs to change some graphic or sounds or something else, project manager is responsible to arrange other teams and solve the problem during time table of testing.

Problems in game mostly occurred on:

1. Logic

2. Other issues

These problems arise from programming logic and the cause is related to CPU, process between hard wares and operating system. These are some examples related to this topic:

- Enemy units are not destroyed after shooting.
- Economy of game is not working true.
- Sounds are not synchronized based on movements and scenes.
- There is conflict between speed and frame rate of game
- Enemy is removed by unknown reason.

3. Graphic

These problems mostly occurred because of library files in operating system related to graphic hard ware and amount of memory which is needed by game engine for graphic items. Sometimes, game setting can cover these problems and sometimes can not. These are some examples related to this topic:

- Sky is rendered with unusual colours.
- Parts of 3D objects are not rendered.
- Textures are not matching on the objects coordination.
- Objects have problems on the edges of screen or levels of game.
- Game screen is bright or dark in unusual way.

4. Sound

Volume synchronization, channel mix and noise are most common problems in sound. Also when an I/O device is work, sometimes sounds have problems.

5. Network

The most important problem on network is none-synchronization between players on network. This issue can seriously affect the business plan of game, especially in first person game style. Players found game boring and none logical during play time. Other problem is lake of send and receiving game data through network. Defining a unique standard for players on network in cases of speed and performance are suggested solution for these problems.

6. Other issues

Game need some input key commands from players like keyboard, joystick or game pad. These inputs must work on time and show results clear. Data items like texts and languages must be prepared without errors as concepts and visualization. If players try to discover game in other way, there must be some anticipation for this reason. For example in simulation game, player tries to fly to other parts of map and do not follow the game objective. Here players must support by special purpose functions which supports this issue during game play. On strategy games, players must be free to command, build and control game items as any styles that they thinks. Controlling chaos in game, when there are lots of items on the stage, is another issue which can make some errors on game.

At the end of test period, game is finalized and prepare for lunch. After game is lunched there can be some hidden problem related to new hard wares that is covered by releasing new patches for game by game providers and this is based on project manager’s long term game business plane. These free patches could be prepared through digital Medias like internet for players. Also after lunch there are some feedbacks like maps, articles and artworks from players which can be used for further developments.

Conclusion

Testers are the first series of players that play the game and it is useful that other team members join testers in testing periods. Also they can get new ideas for next versions or additional patch for game and that is important as game long range business plan.

Appendix A: Examples

Example One:

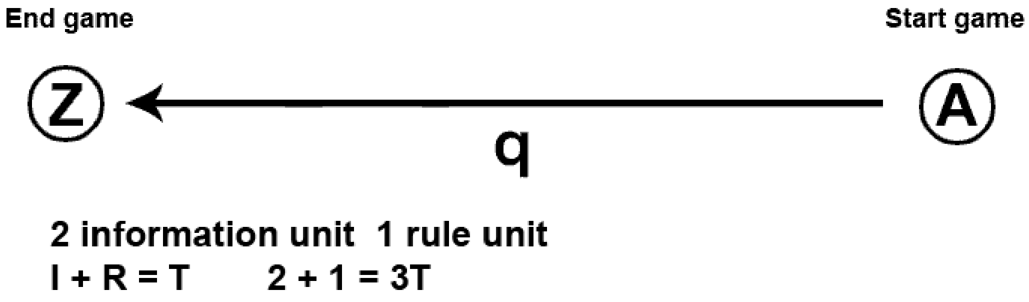


Figure S1-07

Story: Player is an archer. He must shoot at some apples. When he targets all the apples, game will end.

Description:

A: This is the main stage of game.

q: This is a name for a function which controls shooting an arrows. It could be a direct shoot to target (apple) or in-direct shoot to target. (see figure S1-8)

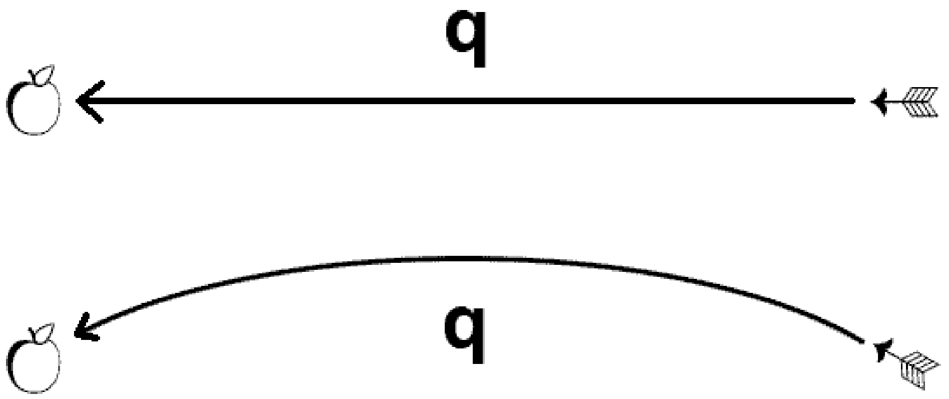


Figure S1-8

Z: This is an ending stage of game. It can be an image for records of high marks or a special movie.

Example Two:

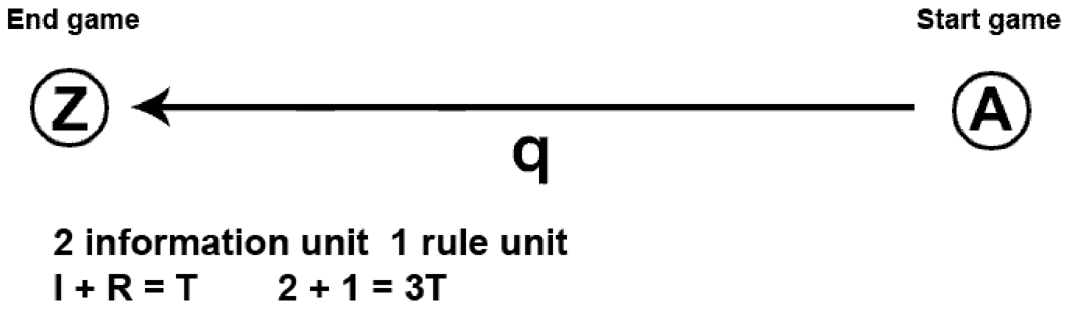


Figure S1-9

Story: A small grey squirrel is passing from a dangerous mountain and fights with some bad animals and reaches to his house.

Description:

A: First place of small grey squirrel.

q: Methods of move and fight for small grey squirrel with enemies.

Z: The home and ending of game.

Example Three:

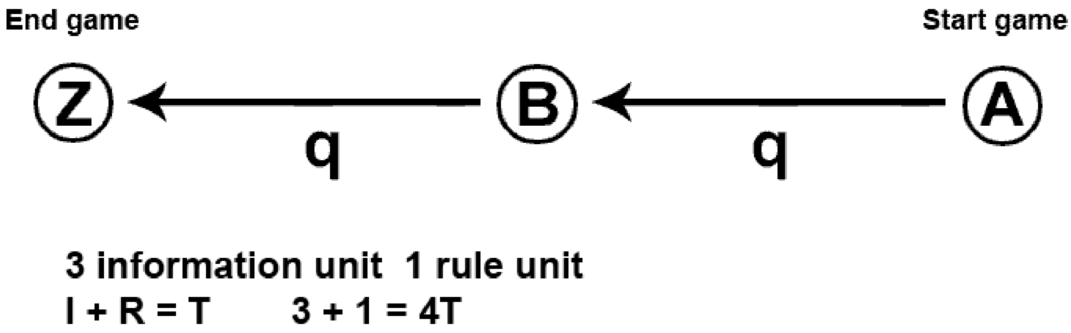


Figure S1-10

Story: A space ship move throw a path to reach to it’s headquarter. It fights with enemies during the game.

Description:

A: Start point of game with selecting various models of ships and weapons.

q: Methods of attacks and destroying enemies.

B: Environment that space ship fights inside that.

Z: Headquarter

Note: “**B**” is an information unit that describe an environment of fight and attack. Before and after fight, all methods for attack and fight (**q**) are same. Have a look to next example for understanding the differences.

Example Four:

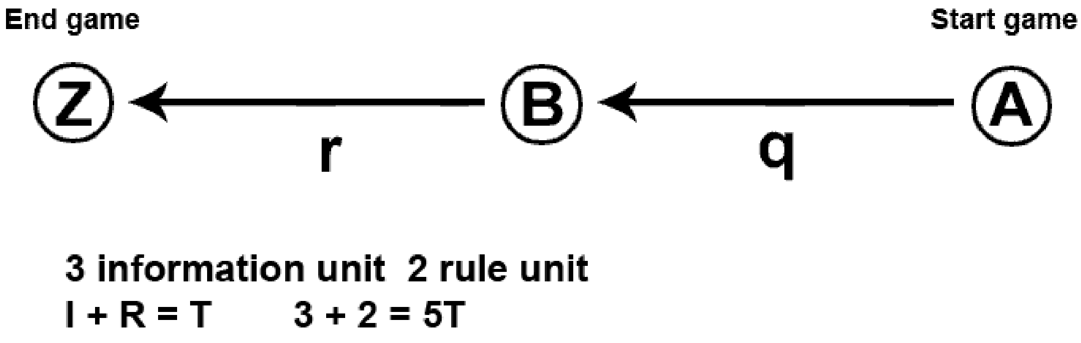


Figure S1-11

Story: A space ship leaves mother ship for bombing a target. It fights with enemies during the way. After bombing he must reach to headquarter, and fighting with much more expert enemies.

Description:

A: Mother ship with full arsenal for space ship.

B: Fight area

q: Methods of moving space ship with bomb, and fighting with enemies.

r: Methods of moving space ship after dropping bomb. It moves faster. Enemy units also have more shooting accessories.

Z: Headquarter

Note: Based on "**Nibok**" method, first we draw logical graph of game and then we adapted the real story based on this graph. We can add more detail to the graph if the story needs more. Adaptation between story and logical graph guarantied the speed of creating game on the next steps of **Nibok** methods, so the story and the logical graph must cover each other perfectly. This prepares a transparency between real story and digital version of that (computer game).

Example Five:

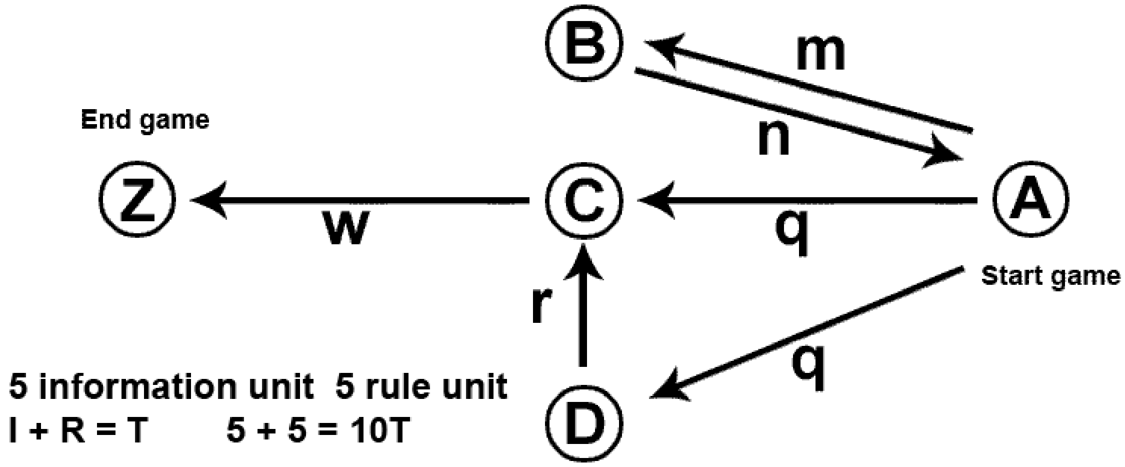


Figure S1-12

Story: A boy in a dream land wants to bring for his little sister some cacao and jam cockies. For buying these items he must collect golden starts, and then give golden starts to kind old mans that prepare the cookies and get the cookies. The golden starts are in hands of a bad witch. Their shapes are circular and sharp and both of them are ok for getting cookies.

Description:

A: Start the game in a dream land. There is a little girl who is sad.

B: Castle of bad witch.

m: Methods of moving to the castle and fighting with enemies for catching golden stars.

n: Methods of moving back from the castle with golden stars and fighting with enemies.

C: The kind old man`s cooking shop which accept golden stars and give cookies.

D: The kind old man`s cooking shop which accept golden stars and give cookies two times more.

q: Methods of moving to cooking shop through some traps that created by bad witch.

r: If boy buys cookies in **D** then there is a special enemy to attack him.

w: Methods of moving to home, Here the game difficulty is high.

Z: End of game and a happy little girl.

- If player set game difficulty to “Hard” then “**r**” will assign to game. In “Easy” mode it should be replaced by “**q**”.

Example Six:

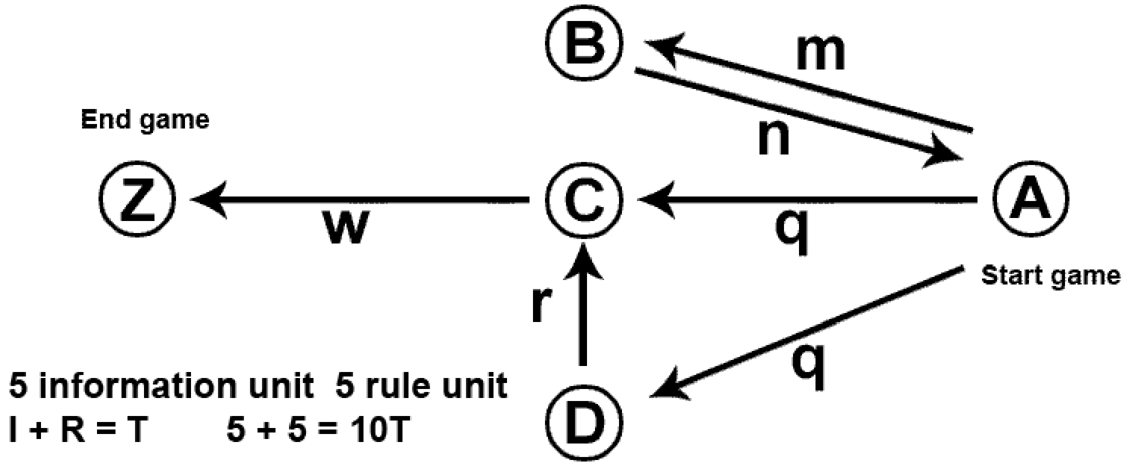


Figure S1-13

Story: A business company which sell farm product, need more production to bye modern farming tools. At the beginning of game you have 10 hectares of farm and $1000 credits. You must produced more product and increase your credits. Then bye modern farming tools and increases your farming production. The goal is reaching to $100,000 credits.
A: Small farm with $1000 credits.
B: Farm with separator grain factory for farm's products.
m: Farming methods
n: Farming methods by using factory
q: Sending products to shopping centre by normal transporting system.
D: Shopping centre A: Local and small
C: Shopping centre B: Far and big
r: If there is not a good business in shopping centre A, we can send products to shopping centre B. Other farmer's business and natural disasters can affect the shopping centre A.
w: Upgrade transporting tools and farming tools
Z: End game with collecting $100,000.

Example Seven:

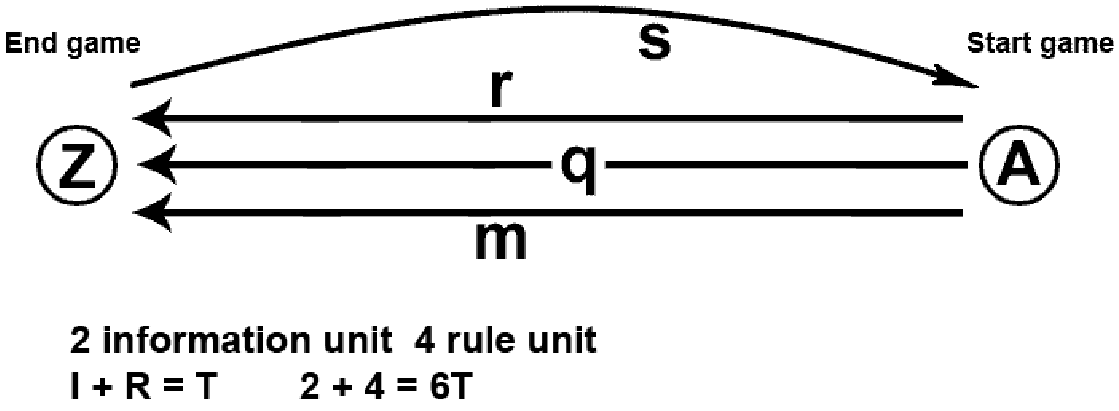


Figure S1- 14

Story: You navigate a Boeing 747 between two airports. This is a simulation game and contains different weather situation as level of difficulty for players.

Description:

A: Starting airport

Z: Destination airport

q: Normal weather's methods

r: Special weather's method

m: in this situation, your plane has some problem in navigation.

s: The pilot is ranked and he can play game continually for earn higher ranks and increase his expertise.

Note: **A** & **Z** can be random locations but their information is absolutely same. Methods for **q**, **r** and **m** could be dynamically changed during the game period based on player experiences.