Designer: Glen Gougeon
Date: 9-20-20

Game: A Nap in Time:
A History Major's Unexpected Journey

Introduction

The Pitch:
Imagine you are a senior level undergraduate History Major. You've already completed your major requirements, but you need to make a unit quota in order to graduate. You picked a few general education classes that seemed interesting, but turned out to be painfully boring. You doze off to sleep. Awaking to the past, you must use all of your wits to escape the daydream, and get back to the present!

THE GAME:

Style.
A Nap in Time is a text-based, historical action-adventure, escape room style game. Ten levels span from the beginnings of the French Revolution in 1789, until the final year of World War II. This single player game, has the user manipulate menus, pickup weapons, documents, and food in order to accomplish missions. The user doesn't know which items might be critical to escaping, and which ones are not. The player has to figure that out as they go. Two recurring menus, and explicit instructions inform the user how to proceed. There are thirteen possible losing scenarios but only four winning ones. There are a hundred possible rounds called heartbeats, before the player loses automatically.

Flow.
The title screen comes up as discussed, the user selects an object from the desk, then begins navigating through the levels. Depending on what they choose from the command menu will determine their next steps. The Action command enters the mission for that level. On-screen instructions are very straight-forward, and are typically the ENTER key, and YES or NO responses as discussed. There are copious prompts to let the user know what to do next.

Beginning of the Game.
An opening title, pitch and selection prompt is the first output of the game. Output is designed to make it clear what the user should do at every step. Following the user's choice to play, the game explains that you fell asleep, but begin to wake up. Your blurred vision clears up, and you see four objects. Selecting one of them will lead the user into the command menu. From there the player has 100 turns to win, or they are stuck in the past forever. The 'heart' variable displayed is the number of heartbeats you have left.

Designer: Glen Gougeon
Date: 9-20-20

Game: A Nap in Time:
A History Major's Unexpected Journey

```
_____________________________________________________________________

      - ~ WELCOME TO: GLEN GOUGEON's
          A Nap in Time :
              A History Major's Unexpected Journey ~ -
_____________________________________________________________________

You are an undergrad History major in your senior year. All that remains for you is
to fill out units in an effort to meet the graduation requirements, but are hopelessly,
painfully, bored by the lecture. It is probably the worst GE course ever devised by man.
Zoning out, you begin to nod off, but awaken in the past. "WAKE UP" you order yourself,
but it is no good. You are stuck in a dream. The only way out is forward. You must explore
the past to get back to the future. What will you do?

You must CHOOSE:

To Play: press <y|Y> + <ENTER>
To Quit: press <n|N> + <ENTER>

=>
```

```
You're head is swimming..
..and everything is a little fuzzy.. but..
You slowly start to see a desk.

<ENTER>
```

The Desk Objects Selection.

```
Choose an object: + <ENTER>

1. aviator goggles
2. a flask
3. .45 ACP casing
4. fabric scrap with Fleurs-de-lis

>
```

Next Steps.
After choosing a desk object, the player is transported to a level associated with the chosen object. The Command Menu will be the next thing they see. This is the recurring game menu that the user will interact with as described below.

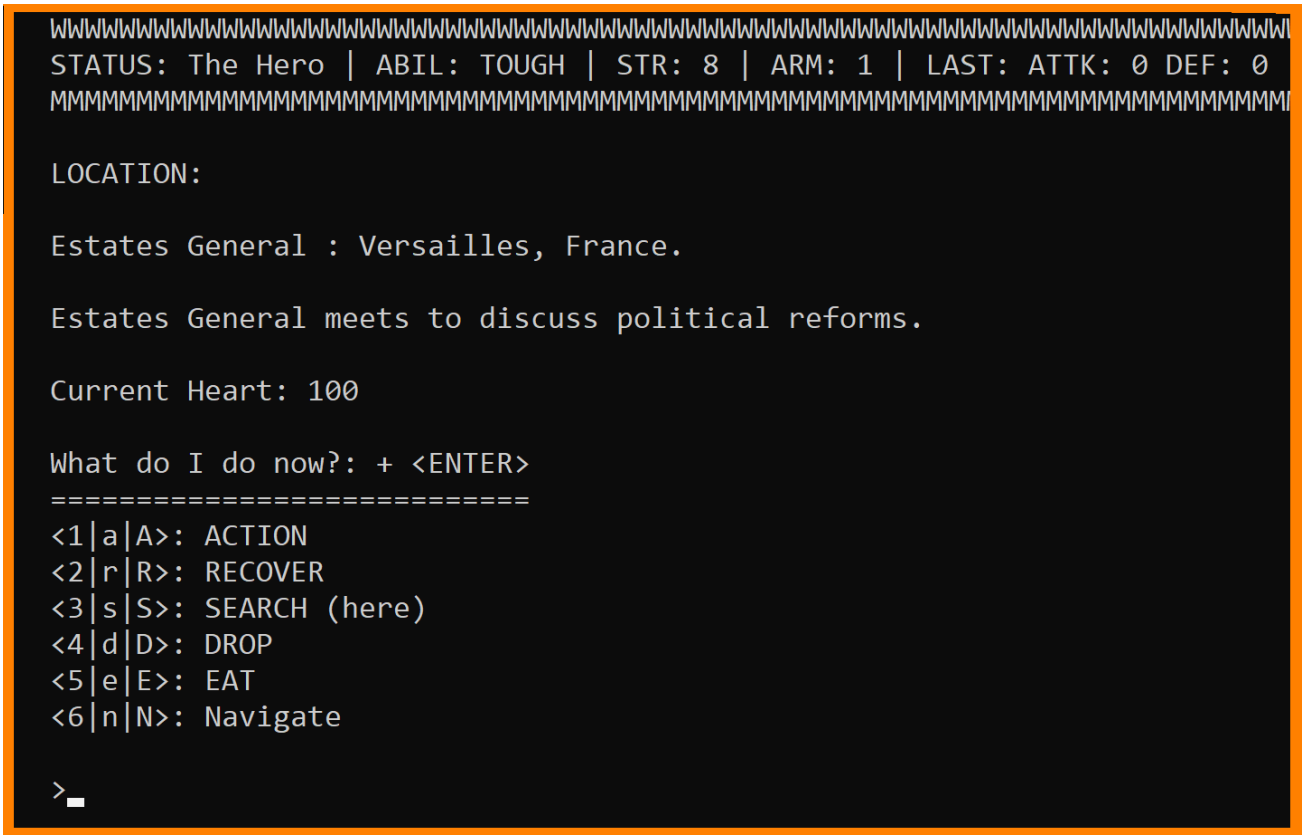
Designer: Glen Gougeon
Date: 9-20-20

Game: A Nap in Time:
A History Major's Unexpected Journey

Command Menu.
(not displayed everywhere)

The primary menu for executing commands. Header information is updated for every executed command including:

- Statistics for *The Hero*, the user's main Game Character
- Location (name of Level)
- Level Description (if first time entering)
- List Command Options



The player can do 1-of-6 things every turn (heartbeat).

- **ACTION**
- **RECOVER**
- **SEARCH**
- **DROP**
- **EAT**
- **NAVIGATE**

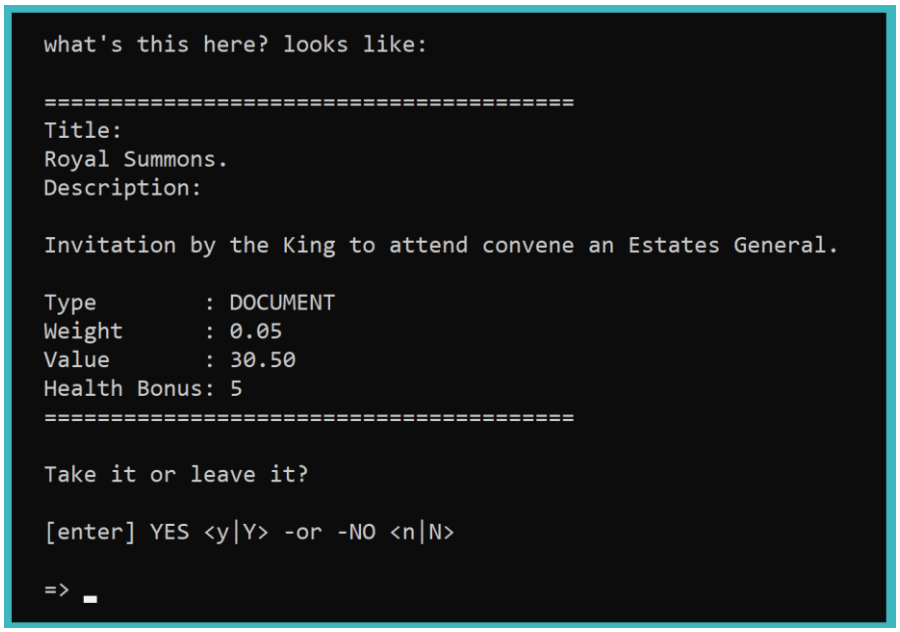
Designer: Glen Gougeon
Date: 9-20-20

Game: A Nap in Time:
A History Major's Unexpected Journey

Action.
Enters the mission sequence for the current level. Some levels allow re-entry for secondary missions, or to navigate away to get certain objects. Missions that require specific items before proceeding will have the player do this. Repeating a mission is possible for some missions, while others execute only once. That is part of the fun.

Recover.
Recover an item left at that level. The user can drop off items that they have in their satchel (queue) at any level of their choosing. If the user's main Game Character discovers an item that they do not want, it is automatically sent to the recovery queue for later retrieval.

Search.
Search for a new item at this level. Items with historical contexts are not necessarily restricted to specific levels. For example, you may find and use a modern weapon from WWII on any level. Searches may result in finding THINGS, WEAPONS, HEAVY-WEAPONS, FOOD, BOMBS, or DOCUMENTS. Undesired items go in that Level's queue for possible recovery.

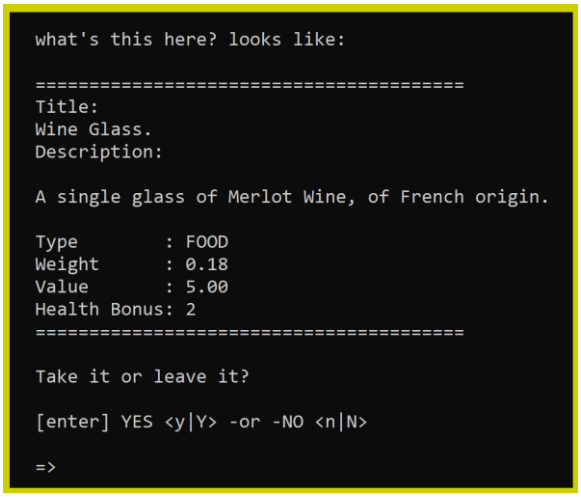


Drop.
Drop off an item on this level. New search items go here automatically if you don't want them for later retrieval. This is a Queue data structure.

Designer: Glen Gougeon
Date: 9-20-20

Game: A Nap in Time:
A History Major's Unexpected Journey

Eat.
Who doesn't like eating? Eating increases the strength parameter of a character. This value will be updated in the status bar of the action menu. Eating adds the health bonus associated with a food to a character's strength, but requires food from your satchel (queue). The advantage of this is that the character can eat anywhere in the game, such as in the middle of a complete mission.



Navigate.
Navigate to another level. A special navigation menu displays a legend of navigation key instructions, and the levels adjacent to the user's current location. The name levels are displayed in the navigation menu only if you have been there before, with the exception of the Bureau of Alcohol office, since that is where you woke up at the beginning of the game.

Eras and Levels.
There are four historical eras in the game that include the early French Revolution, Prohibition and the American Mafia, Early Flight with Charles Lindbergh, and Allied Special Operations during World War II. Every level is both a specific location, and historical event, though some are less specific, but ample description is provided. The location name, date, and historical significance are available at the beginning of every level with the ACTION command.

Historical Events.
Each game level is a reference to one of ten historical events including:
French Revolution.

- May 5, 1789: Meeting of the Estates General, Versailles, France.
  The political beginning of the French Revolution.

- July 14, 1789: Storming of the Bastille, Paris, France.
  The first significant violence of the French Revolution.

- July 1789: The Great Fear, French Countryside.
  Abolition of Noble Privileges provided by Fear and Force.

- October 5, 1789: The October Days, Versailles, France.
  Parisian women subdue the Monarch and remove them to Paris.

- June 21, 1791: The Flight to Montmedy, France. Capture at Varennes.
  Failed escape attempt by the King to regain power.

Designer: Glen Gougeon
Date: 9-20-20

Game: A Nap in Time:
A History Major's Unexpected Journey

Early Flight.

- May 21, 1927. Charles Lindbergh makes the first non-stop solo flight from New York to Paris.

Prohibition Era.

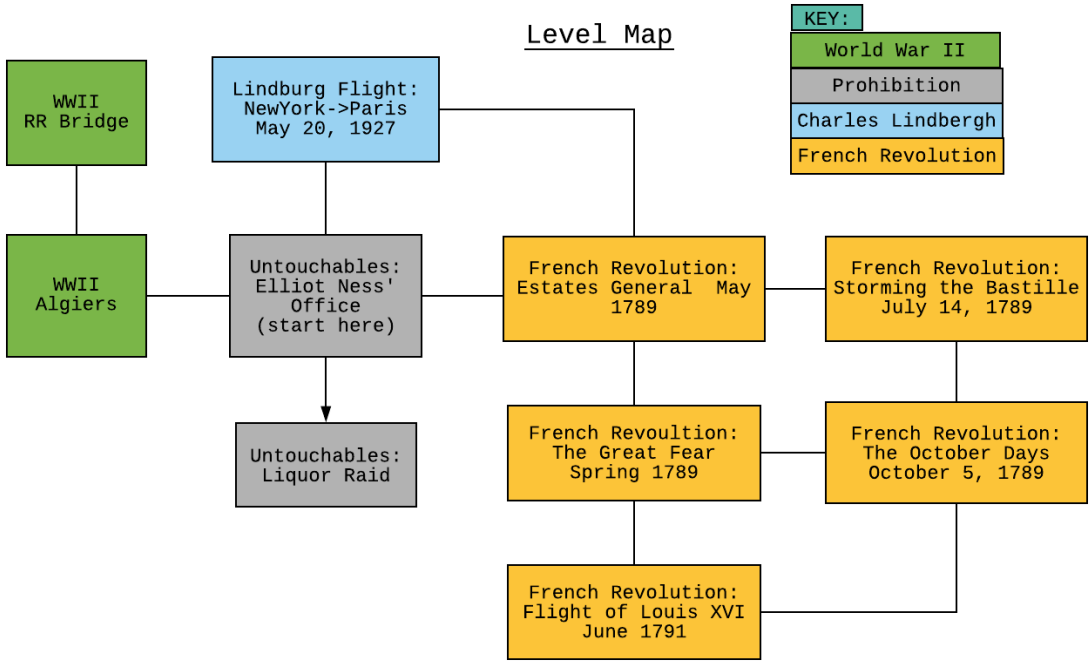
- June 16, 1931: Evidence against Alfonse Capone pled guilty to Tax Evasion.
  Evidence provided by agents of the Treasury Dept. Scored a long awaited conviction of the notorious mafia boss.

- Circa June 1930. Elliot Ness and the Untouchables Capone's Liquor operations.
  Pressure and discovery of Capone were had by raiding his operations by way of the Bureau of Alcohol.

WWII.

- July 31, 1944. Special Operations Jedburgh units drop into occupied Southern France.
  Three-man teams performed training of French resistance Partisans ahead of invasion.

- August 12, 1944. Jedburgh teams and Partisans sabotage Germans ahead of Operation Dragoon.
  August 15, 1944 saw the invasion of the French southlands. Jedburghs created mayhem ahead of that invasion such as attacking railroad infrastructures.



Designer: Glen Gougeon
Date: 9-20-20

Game: A Nap in Time:
A History Major's Unexpected Journey

VICTORY MISSIONS.
Spoiler alert. There are four missions that let you win the game. There are plenty of ways to die, thirteen in all. You can pick your flavor of victory.

Revolutionary:

Support for the revolution required obvious significant political change. Violence, and paper-work required. This game uses a lot of paperwork, special **document** items that can help you escape such as, the Declaration of Rights of Man and of the Citizen, the August Decrees, and the King's official seal of approval. This is the most difficult way to win. The Declaration for instance, can only be had by burning down the Chateau in the October Days Level.

Loyalist:

The longest route to victory, requires escorting the King and royal family to safety. You must re-enact the failed escape to the fortress of Montmedy near Austria without being captured. Some initial preparatory items are required, and will not let you continue without them. Watch what you say, and escape is certain!

Conviction:

The fasted way to WIN and to DIE! Get a warrant, get armed, and fight the bad guys. You must successfully attack one of Capone's Liquor operations without being killed by his thugs. There is something you'll want if you survive. Use that to win the game.

Sabotage:

Blow something up! Well, a railroad-bridge specifically. You'll need a suitcase of TNT to get the job done, and you can only steal it from the Nazis. Fight them to get what you need, then go next door and blow up their bridge. WARNING Explosives are dangerous!

Items.
Each level contains items that can be categorized as **things**, **weapons**, **heavy weapons**, **bombs**, **documents** and **food**. Items are placed both specifically and at random. Having needed items in easy to find places is common but not the rule. Explosives for example, can only be gained by fighting the Nazis after jumping into France. Items are what you **recover**, **search** for, and **drop** on a level.

Designer: Glen Gougeon
Date: 9-20-20

Game: A Nap in Time:
A History Major's Unexpected Journey

Carrying Items.
Every character in the game has a pocket that can hold a single item, and a Satchel (queue) which can hold several items. Item weights are part of the game visuals only. They do not limit the carrying ability of the characters.

ITEMS. ***There are 30 different items in the game**
Every Item has the following members:
•Title •Description •Type •Weight •Value •Health-Bonus •Power

Item Type	List
Things	Aviator Hat, Goggles, Emergency Kit, Torch, Fedora, Concordance Change of Clothes
Documents	Royal Seal, Royal Summons, Warrant, Capone's Ledger, August Decrees, Declaration of Rights, De Bouille's Map, Phony Passports
Weapons	Pike, Sabre, Charleville, Smith, Shotgun, Thompson
Heavy Weapons	Artillery, Round-Shot
Bomb	Exploders
Food	Wine Glass, Stale Bread, Croutons, Hamburger, Whiskey, Bugs

CHARCTERS.
There are eleven character (sub-classes) in the game including "The Hero" (user).

The Hero

Ability: TOUGH
Strength: 8
Armor: 1

Every character can have a:

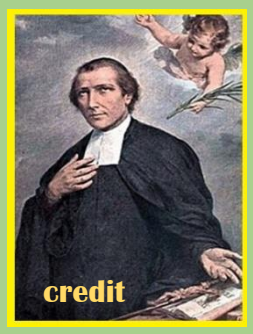
- Name
- Ability
- Strength
- Armor
- Last Attack Roll
- Last Defense Roll
- Pocket
- Satchel

Henchmen

Ability: TOUGH
Strength: 8

Designer: Glen Gougeon
Date: 9-20-20

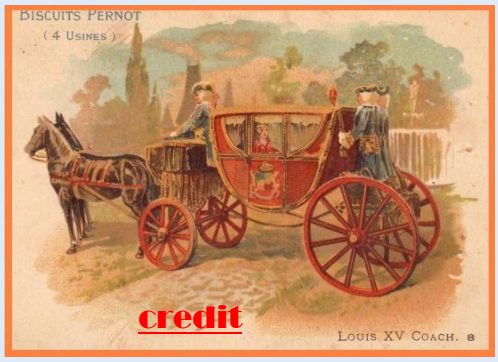
Game: A Nap in Time:
A History Major's Unexpected Journey

Louis XVI  Ability: RUTHLESS Strength: 6	Priest  Ability: PRAYER Strength: 20	Marie Antoinette  Ability: CHARISMA Strength: 4
Noblemen of the Sword  Ability: TOUGH Strength: 8 Armor: 3	Housewife  Ability: PRAYER Strength: 10	Bourgeoisie  Ability: CHARISMA Strength: 6
Palace Guards  Ability: DUMB Strength: 9 Armor: 4	Peasants  Ability: DUMB Strength: 7	Nazis  Ability: DUMB Strength: 8

Designer: Glen Gougeon
Date: 9-20-20

Game: A Nap in Time:
A History Major's Unexpected Journey

VEHICLES:
There are four vehicles in the game namely, a carriage, a fishing boat, a food cart, and the Spirit of Saint Louis, Charles Lindbergh's seminal aircraft. Every vehicle is a child of the base class **Vehicle**. Every vehicle therefore inherits the name, Original Equipment Manufacturer (O.E.M.), classification, crew size, pilot, speed, weight, and value members from the base class.

Carriage  	The Royal Carriage of the Bourbon Monarch and family. Seating specifically for each member of the family, the nanny, and two palace guards. Used during the escape to Montmedy. Name: Royal Carriage Classification: TRANSPORT Crew Size: 4 (including pilot) Passenger Size: 4
Fishing Boat  	Eastern-Rig type fishing tub. Common fishing tug circa. 1920-1930. Charles Lindbergh spotting a fishing boat when he was approaching the coast of Ireland, of which this may have been the type. Used primarily for looking at.
Food Cart  	Contains bed with 10 items inside. Guarded by the driver. Attempt to attack it, and you will either get the items, receive some damage, or possibly be killed.

Designer: Glen Gougeon
Date: 9-20-20

Game: A Nap in Time:
A History Major's Unexpected Journey

The Spirit of Saint Louis

- **Name**: "Spirit of Saint Louis"
- **OEM**: Ryan Airlines Corporation
- **Model**: Ryan M-2 (modified)
- **Engine**: Wright Whirlwind J-5C
- **HP**: 223
- **Wingspan**: 46 ft.
- **Length**: 27 ft. 8 in.
- **Classification**: TRANSPORT
- **Crew Size**: 1
- **Pilot**: Charles Lindbergh
- **Color**: Silver
- **Speed**: 120 m.p.h.
- **Empty Weight**: 2150 Lbs.
- **Max Loaded Weight**: 5135 Lbs.
- **Fuel**: 450 U.S. gals.
- **Value**: $6,000.00 U.S. (airframe only)

COMBAT:

Character to Character combat is simulated through the rolling of dice. Special game characters have additional rolls for initiative and regenerating health. Generic soldier characters, such as Henchmen for example, do not have any initiative.

Attacking.
Characters roll their attack die (quantity and type vary). The defending game character's special ability may give them and advantage or dis-advantage depending on the aggressor's attack() function setting. The last attack roll of the aggressor is saved as part of their statistics.

Defending.
Characters also roll defensive die during combat. A hierarchy dictates how an incoming attack roll is defended against. The attack roll goes to the defense roll, then the defender's armor, and the defender's strength last. For example:
Attacker rolls: 5, Defender rolls: 4, then the defender would have 5 – 4 = 1 possible damage. Suppose the defender has armor of 3, their armor is reduced by 3 – 1 = 2, and we are done.

Attrition.
A game character that has NEGATIVE strength dies. This includes the main Hero character.

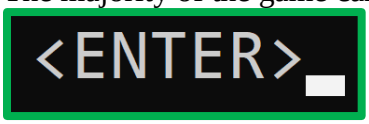
Designer: Glen Gougeon
Date: 9-20-20

Game: A Nap in Time:
A History Major's Unexpected Journey

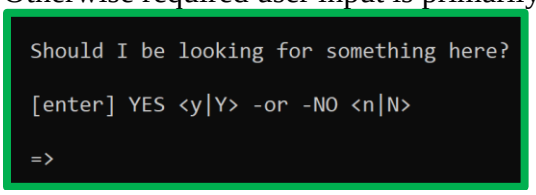
CONTROLS:

Common.

The majority of the game can be played with the **<ENTER>** key.



Otherwise required user input is primarily a yes or no.



Desk Object Selection Controls.
The first selection of desk objects, and the navigation menu actually take the same input values.
The inputs are either: Integers [1..4] or the right-hand characters of the home row: [j, k, l, ;].
This is the same configuration used in the Vi/Vim editor tool for curser navigation.

Direction	up	down	left	right
integer	1	2	3	4
character	j	k	(lower case L) l	(semi-colon) ;

original image credit

original keyboard image credit

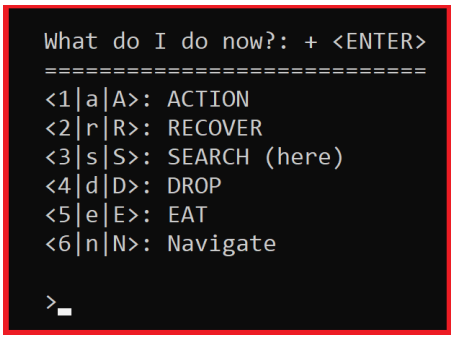
Designer: Glen Gougeon
Date: 9-20-20

Game: A Nap in Time:
A History Major's Unexpected Journey

Navigation Selection Controls.
Functionally the same as Desk Object Selection Controls, but visually different. Valid keys are displayed on the right-hand side.

Command Menu Selection Controls.
The command menu has 6 previously described options. This describes how to select one of those options. Each option selection allows:

Command	integer	lower-case	upper-case
Action	1	a	A
Recover	2	r	R
Search	3	s	S
Drop	4	d	D
Eat	5	e	E
Navigate	6	n	N



Designer: Glen Gougeon
Date: 9-20-20

Game: A Nap in Time:
A History Major's Unexpected Journey

COMMAND USE CASE EXAMPLES:

Action: Start a Mission.
Action scenarios are omitted here for brevity. See Level Architectures for details.

Recover: Pick up something you left behind.
Pick up something stored in the droppedItems Queue from the level you are in.

Use Case: The user didn't drop anything at the level.

The prompt

```
Now where did I leave that thing?? ~~
Do you want to get something you already found here?

[enter] YES <y|Y> -or -NO <n|N>

=>
```

Yes response

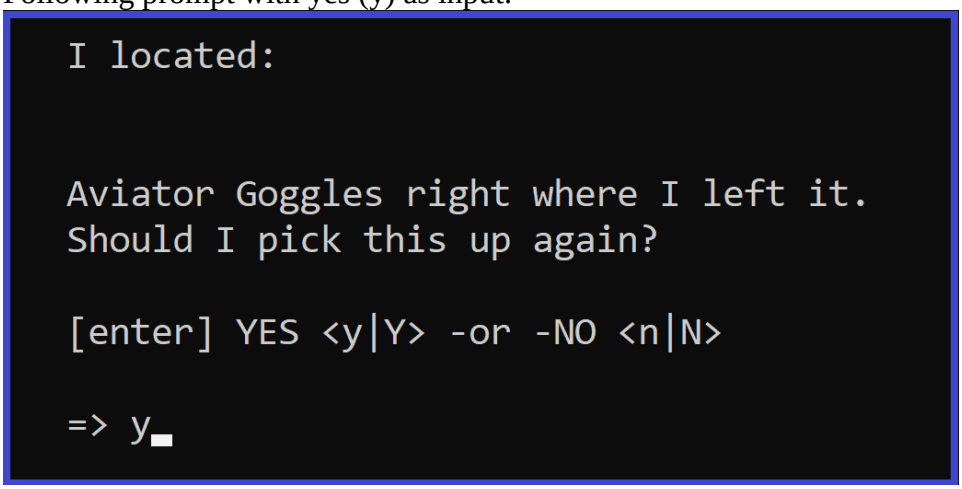
```
Now where did I leave that thing?? ~~
Do you want to get something you already found here?

[enter] YES <y|Y> -or -NO <n|N>

=> y

I thought it was here, but now it's gone.

<ENTER>
```

Designer: Glen Gougeon
Date: 9-20-20

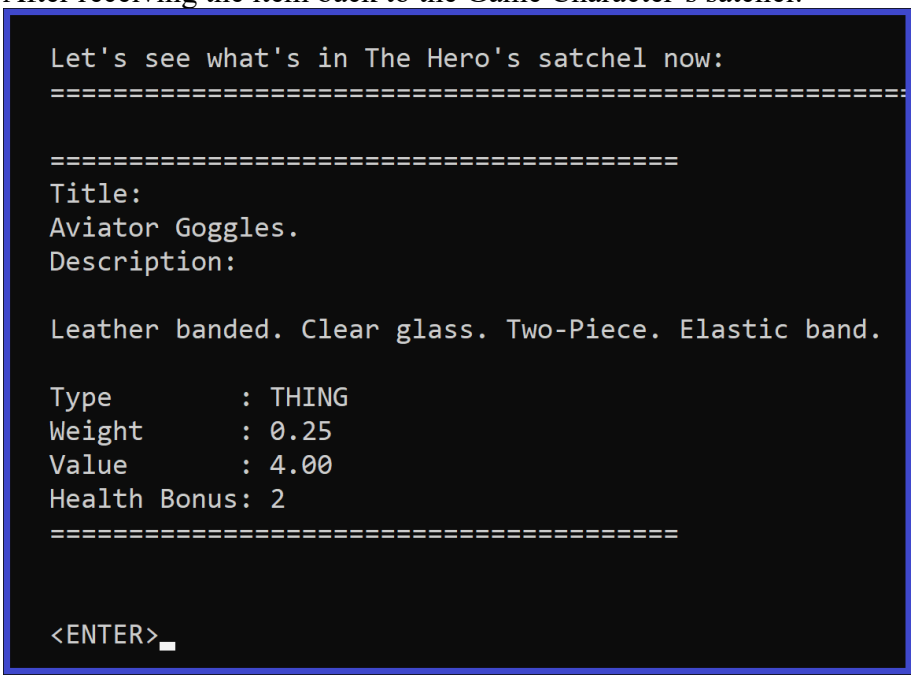
Game: A Nap in Time:
A History Major's Unexpected Journey

Use Case: User did drop something at the level and wants to retrieve it again.

Following prompt with yes (y) as input:



After receiving the item back to the Game Character's satchel:



Designer: Glen Gougeon
Date: 9-20-20

Game: A Nap in Time:
A History Major's Unexpected Journey

Search: Explore for new items.
Use Case: The user wants to search for a new item, and there is something new at on that level.

The prompt.

```
Should I be looking for something here?

[enter] YES <y|Y> -or -NO <n|N>

=>
```

Yes (y) was input, and the revealed item is printed.

```
========================================
Title:
torch.
Description:

Oak branch. Cloth head. Dipped in oil.

Type        : THING
Weight      : 1.15
Value       : 1.00
Health Bonus: 0
========================================

Take it or leave it?

[enter] YES <y|Y> -or -NO <n|N>

=>
```

The user's updated satchel (queue) is then printed.

```
Let's see what's in The Hero's satchel now:
============================================================

========================================
Title:
Aviator Goggles.
Description:

Leather banded. Clear glass. Two-Piece. Elastic band.

Type        : THING
Weight      : 0.25
Value       : 4.00
Health Bonus: 2
========================================

========================================
Title:
torch.
Description:

Oak branch. Cloth head. Dipped in oil.

Type        : THING
Weight      : 1.15
Value       : 1.00
Health Bonus: 0
========================================


<ENTER>
```

Designer: Glen Gougeon
Date: 9-20-20

Game: A Nap in Time:
A History Major's Unexpected Journey

Drop: Leave something where you are now.

Use Case: The user already has an item in their satchel (queue) and drops it off at a new level.

First, the user wants to review what is in their satchel (queue).

```
Do you want to see what's in your satchel first?

[enter] YES <y|Y> -or -NO <n|N>

=> y
```

Then the satchel contents are printed.

```
Satchel contents:
=================

========================================
Title:
Aviator Goggles.
Description:

Leather banded. Clear glass. Two-Piece. Elastic band.

Type        : THING
Weight      : 0.25
Value       : 4.00
Health Bonus: 2
========================================

========================================
Title:
torch.
Description:

Oak branch. Cloth head. Dipped in oil.

Type        : THING
Weight      : 1.15
Value       : 1.00
Health Bonus: 0
========================================


<ENTER>
```

Designer: Glen Gougeon
Date: 9-20-20

Game: A Nap in Time:
A History Major's Unexpected Journey

Another prompt to validate the user indeed wants to drop off an item.

```
Do you still want to drop an item off at this location?

[enter] YES <y|Y> -or -NO <n|N>

=>
```

Since there may be multiple items in the user's satchel, the drop command will have to cycle through every item until the desired item is found. Once entered an item MUST be dropped. It can be picked up again of course through the Recover command in the command menu. This time the user wants to drop off their Aviator Goggles.

```
Is this it? :
=============

Aviator Goggles

[enter] YES <y|Y> -or -NO <n|N>

=> y
```

Another prompt to confirm if they want to drop off more than one item. This time the user doesn't want to.

```
Do you want to leave anything else here?

[enter] YES <y|Y> -or -NO <n|N>

=> n
```

Eat: Eat some food from your satchel.

Use Case: The user has something in their satchel, and they want to eat it. This will give them more strength depending on the health bonus of the food item.

The Prompt

```
Still hungry?

[enter] YES <y|Y> -or- NO <n|N>

=>
```

→

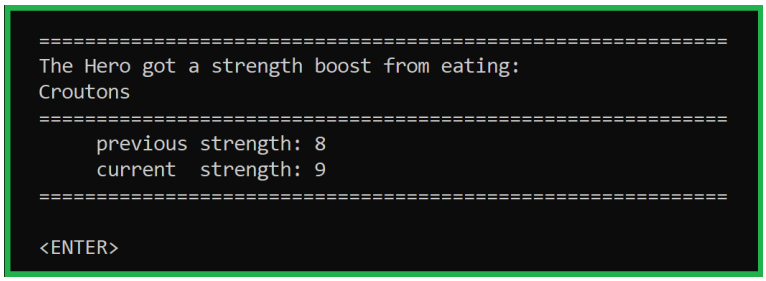
The first food item found will be displayed, and a prompt will ask if the user wants to eat it. Assume the user wants to each these croutons and gain strength of 1 point. (nutritional value)

```
Here's something to eat:
========================

Croutons
Nutritional value: 1
============================

Are you going to eat that?

[enter] YES <y|Y> -or- NO <n|N>

=>y
```

Designer: Glen Gougeon
Date: 9-20-20

Game: A Nap in Time:
A History Major's Unexpected Journey

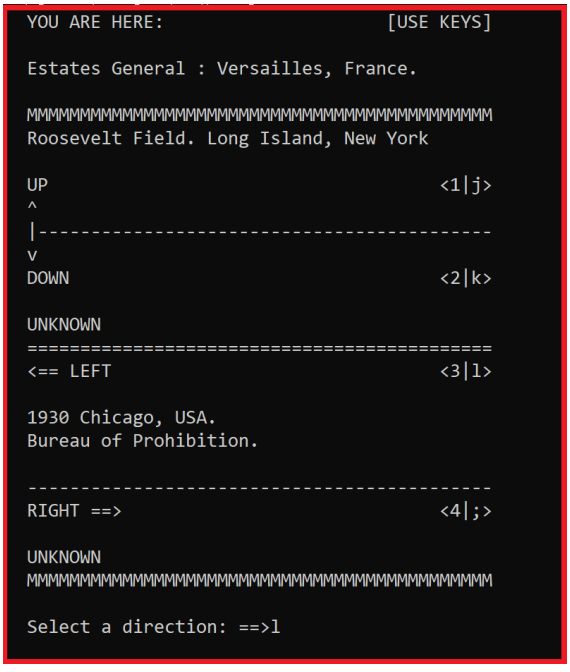
The result of eating the food will be printed before returning to the command menu. The increased strength is also updated in the user's Game Character "The Hero." in the status bar.



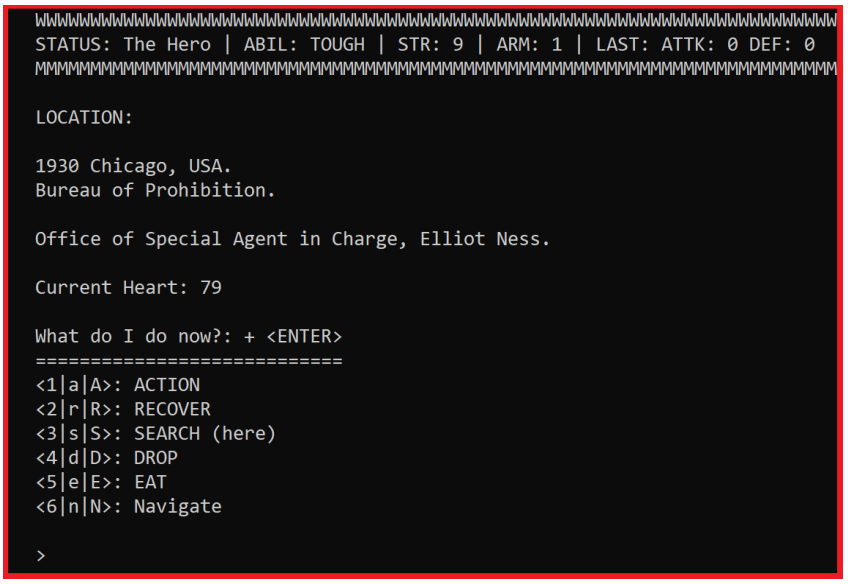
Navigate: Go somewhere else.

Use Case:

The user wants to go to a new location to the left. In this case the Bureau of Prohibition.



The user is brought back to the command menu, but now in the Bureau of Prohibition as desired.



Designer: Glen Gougeon
Date: 9-20-20

Game: A Nap in Time:
A History Major's Unexpected Journey

COMPILATION INSTRUCTIONS:
bold = what to type at the command line

1. Unzip folder.
2. TO COMPILE:
    run the following command to compile the program:
        make main

3. To START the program AFTER compilation run the command:
        main

The game has now started. Follow the screen prompts for game play.

ABORTING:
    To stop at any time during the middle of the program run the command:
        control-C

TO CLEAN:
    Run the following command:
        make clean

DEVELOPMENT

Environment.
Windows 10 and Linux

Tools.
    Microsoft Visual Studio 2017
    Git for Visual Studio
    Vim text editor (on OSU-Flip Linux Servers)
    Putty Terminal Emulator for Server access.
    Microsoft Word (this document)

Servers.
    GitHub
    Flip3 at Oregon State University's Engineering Dept.

Languages.
    C++

Designer: Glen Gougeon
Date: 9-20-20

Game: A Nap in Time:
A History Major's Unexpected Journey

ARCHITECTURE

High Level Overview.

Structurally the game consists of levels instantiated on the heap by the game class. Every level contains four pointers to levels that enable connection to adjacent places on the map. Here map is not the map data structure. Following instantiation, a specific sequence of connections take place to make the correct adjacent connections. The Game Class construction is an abstraction of the map I have described.

The main() function wrapper includes srand() to ensure the pseudorandom generator works properly for the dice used during gameplay, creates a Game on the stack, which in turn builds a map on the heap. The sequence of calls to enter the game shell are:

```
main()
enterPrimaryGameMenu()
        printTitleBlock()
        startGameYesNo()
                menuAfterYes()
                        runGameShell()
```

Within this shell is where the command menu and navigation menu choices execute most of the game play.

The use of pointers, memory allocation, the shell found in runGameShell(), the Stack and Queue data structures were the most important structural components of the game. The level mission structures executed by the Action command were made possible through inheritance. This flexibility worked perfectly to create distinct missions for each level.

A detailed explanation of those architectures is better served by the flow diagrams I have provided below the Level Class Architecture UML and the Character-Vehicle Associative UML.

Designer: Glen Gougeon
Date: 9-20-20

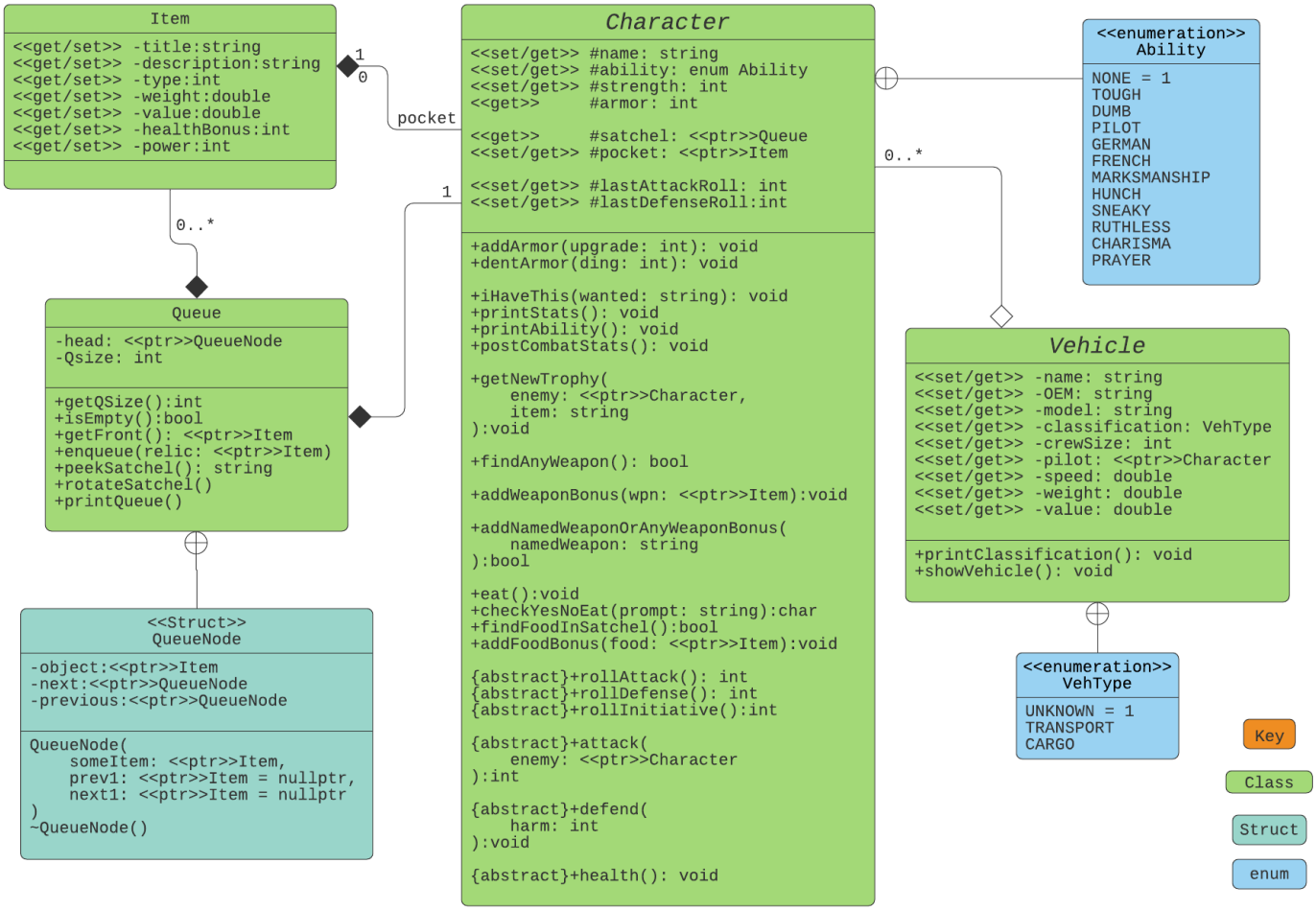
Game: A Nap in Time:
A History Major's Unexpected Journey

Level Class Architecture

Designer: Glen Gougeon
Date: 9-20-20

Game: A Nap in Time:
A History Major's Unexpected Journey

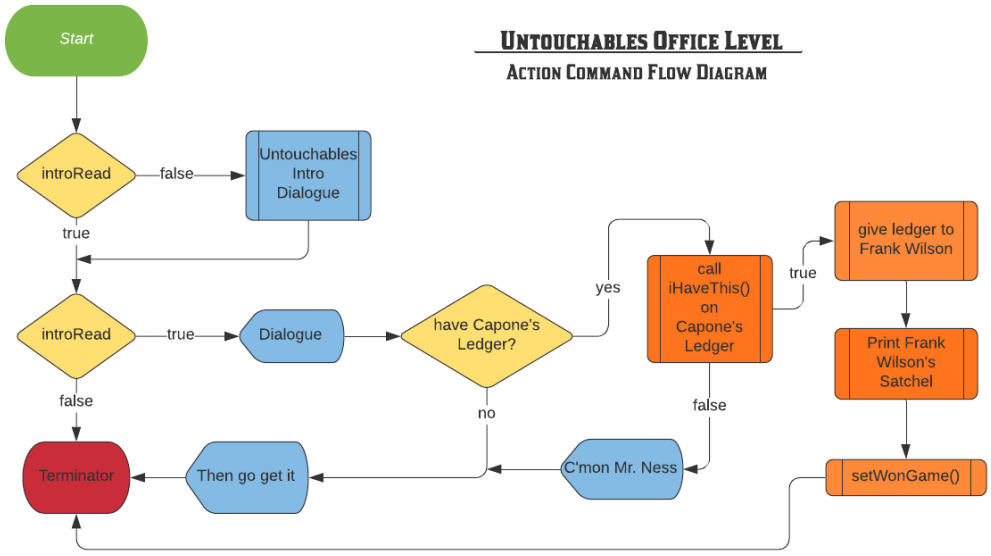
Character-Vehicle Association UML

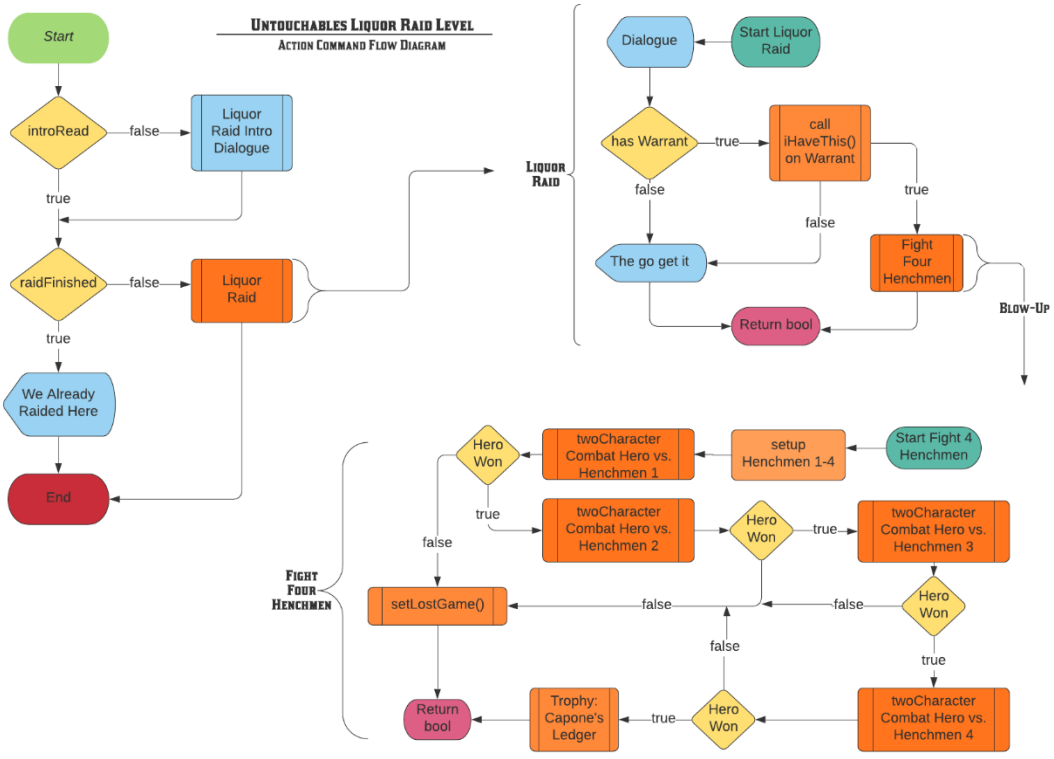


Designer: Glen Gougeon
Date: 9-20-20

Game: A Nap in Time:
A History Major's Unexpected Journey

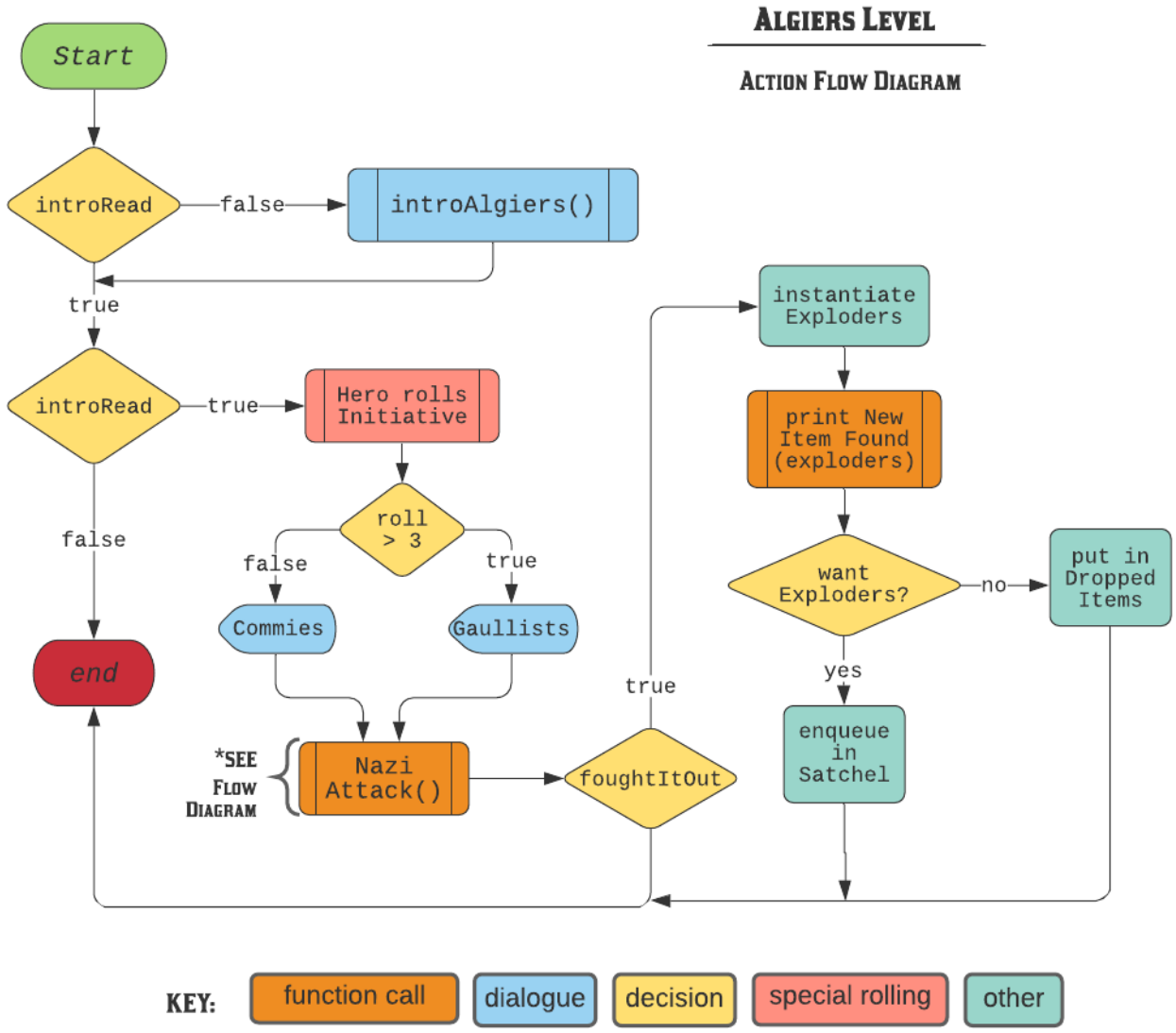
LEVEL LOGIC FLOW DIAGRAMS:





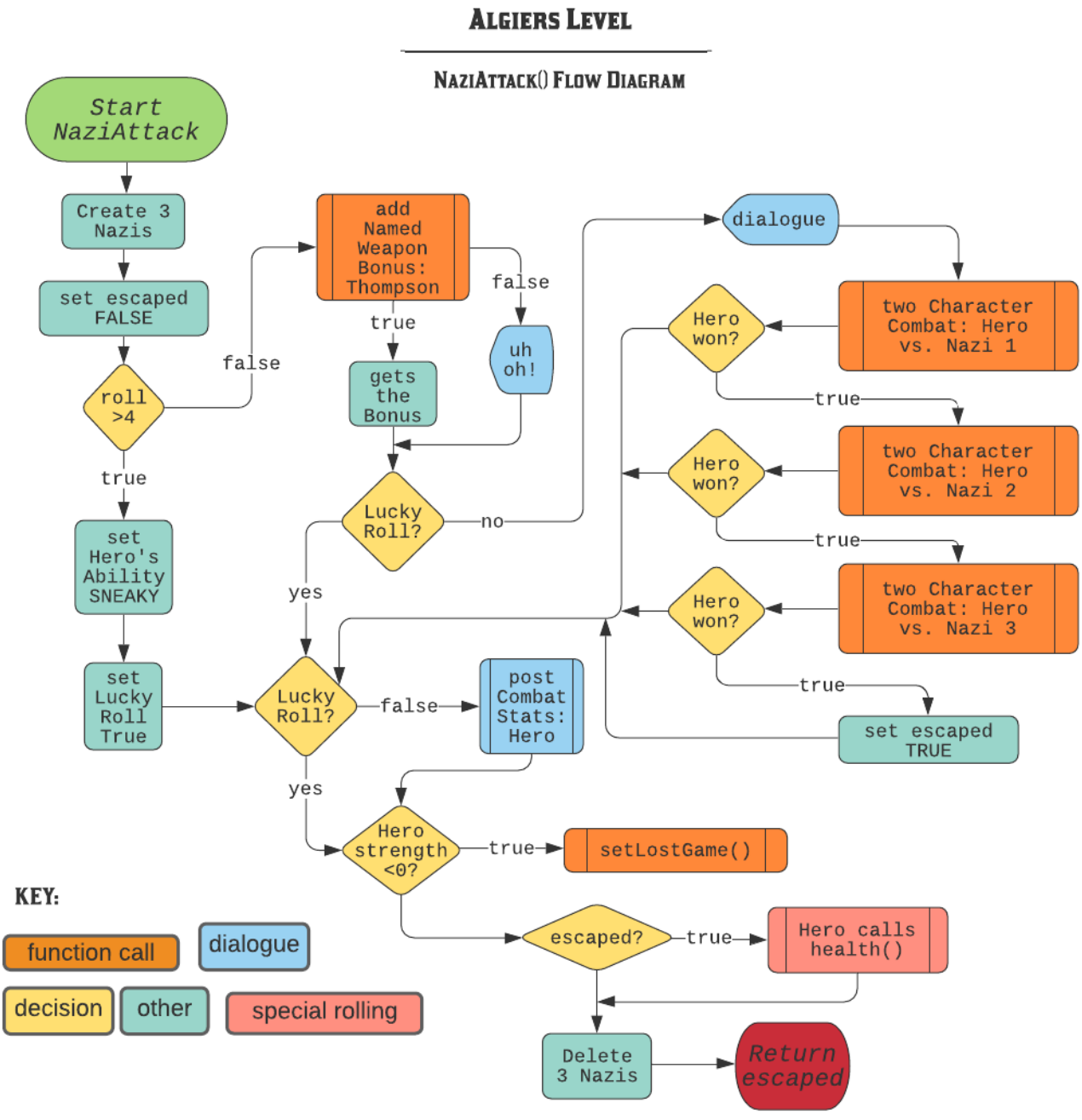
Designer: Glen Gougeon
Date: 9-20-20

Game: A Nap in Time:
A History Major's Unexpected Journey



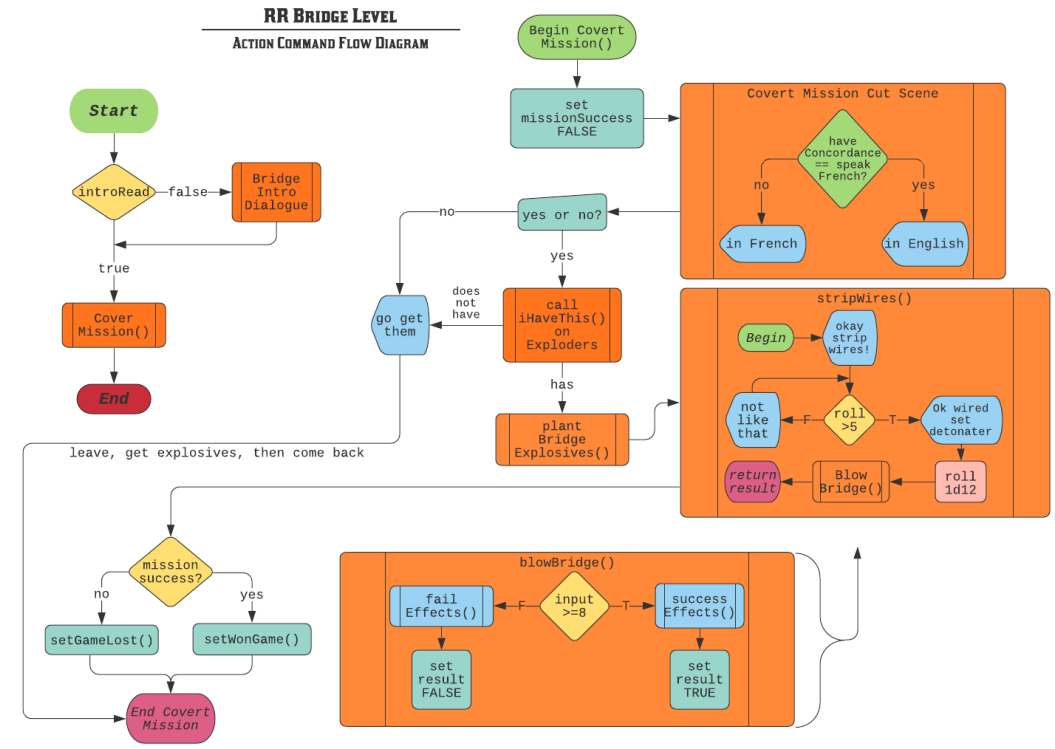
Designer: Glen Gougeon
Date: 9-20-20

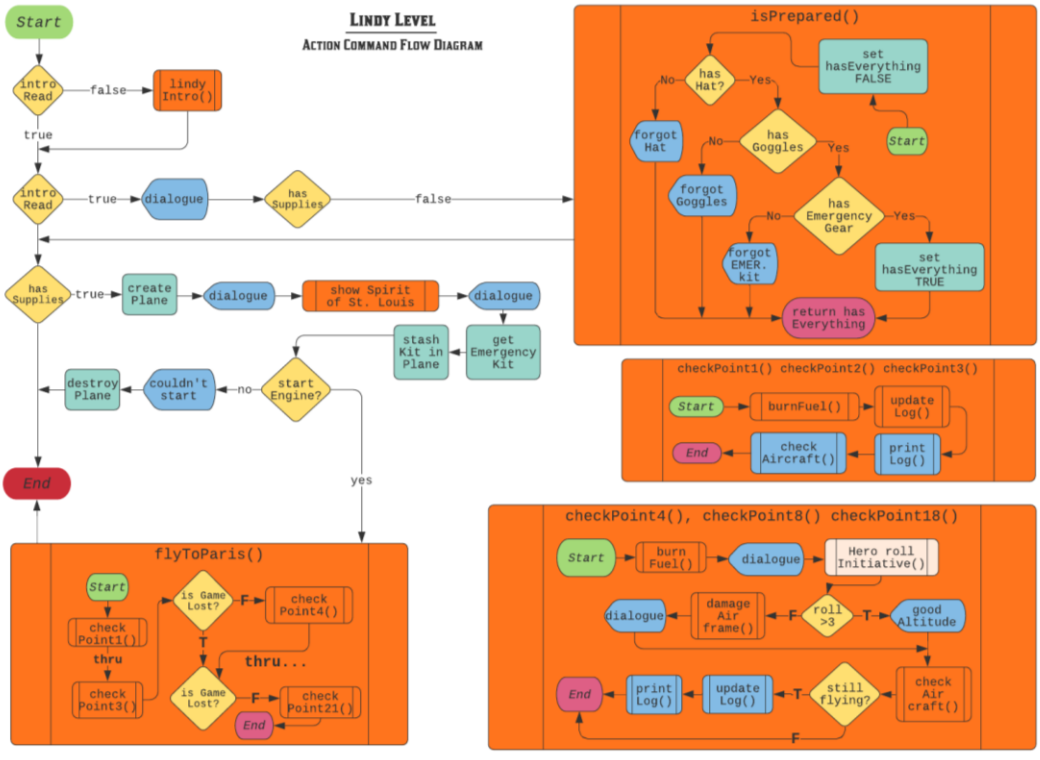
Game: A Nap in Time:
A History Major's Unexpected Journey



Designer: Glen Gougeon
Date: 9-20-20

Game: A Nap in Time:
A History Major's Unexpected Journey



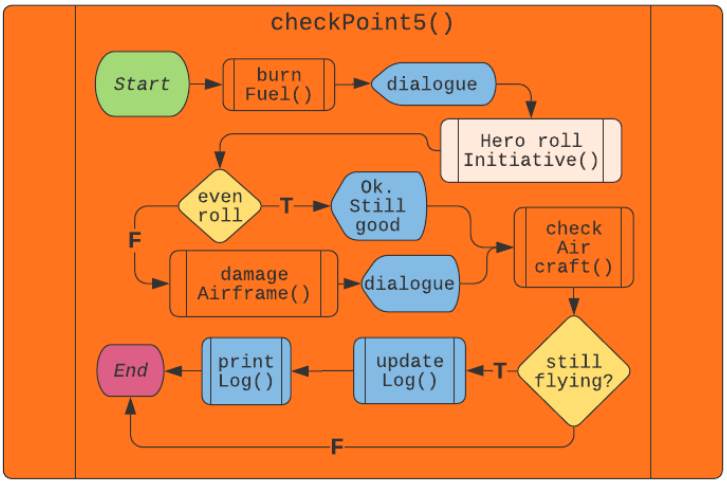


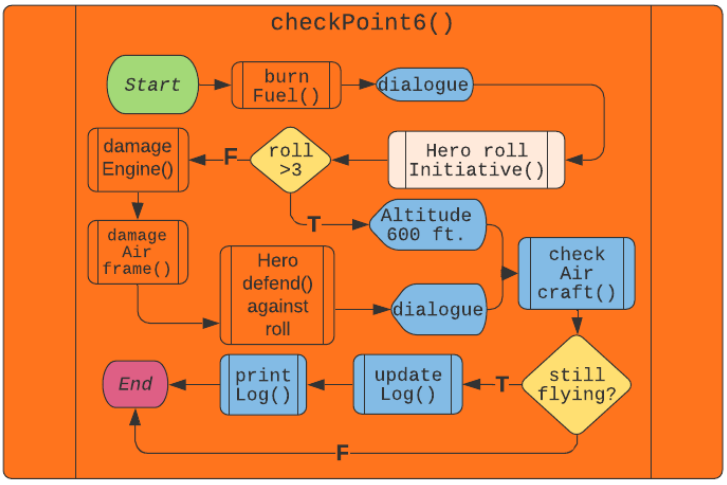
Designer: Glen Gougeon
Date: 9-20-20

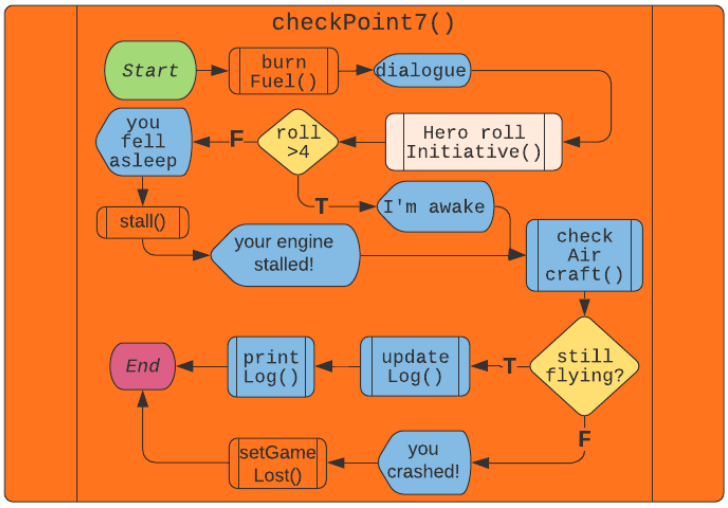
Game: A Nap in Time:
A History Major's Unexpected Journey

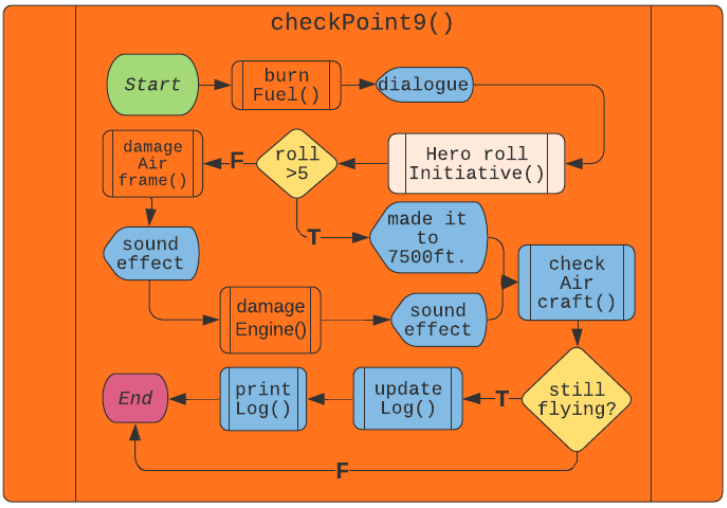
Lindy Level

checkPoint() Flow Diagrams







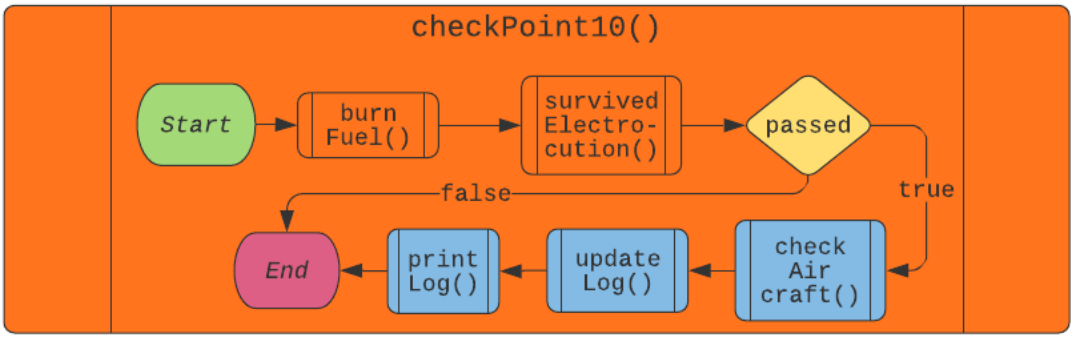


Designer: Glen Gougeon
Date: 9-20-20

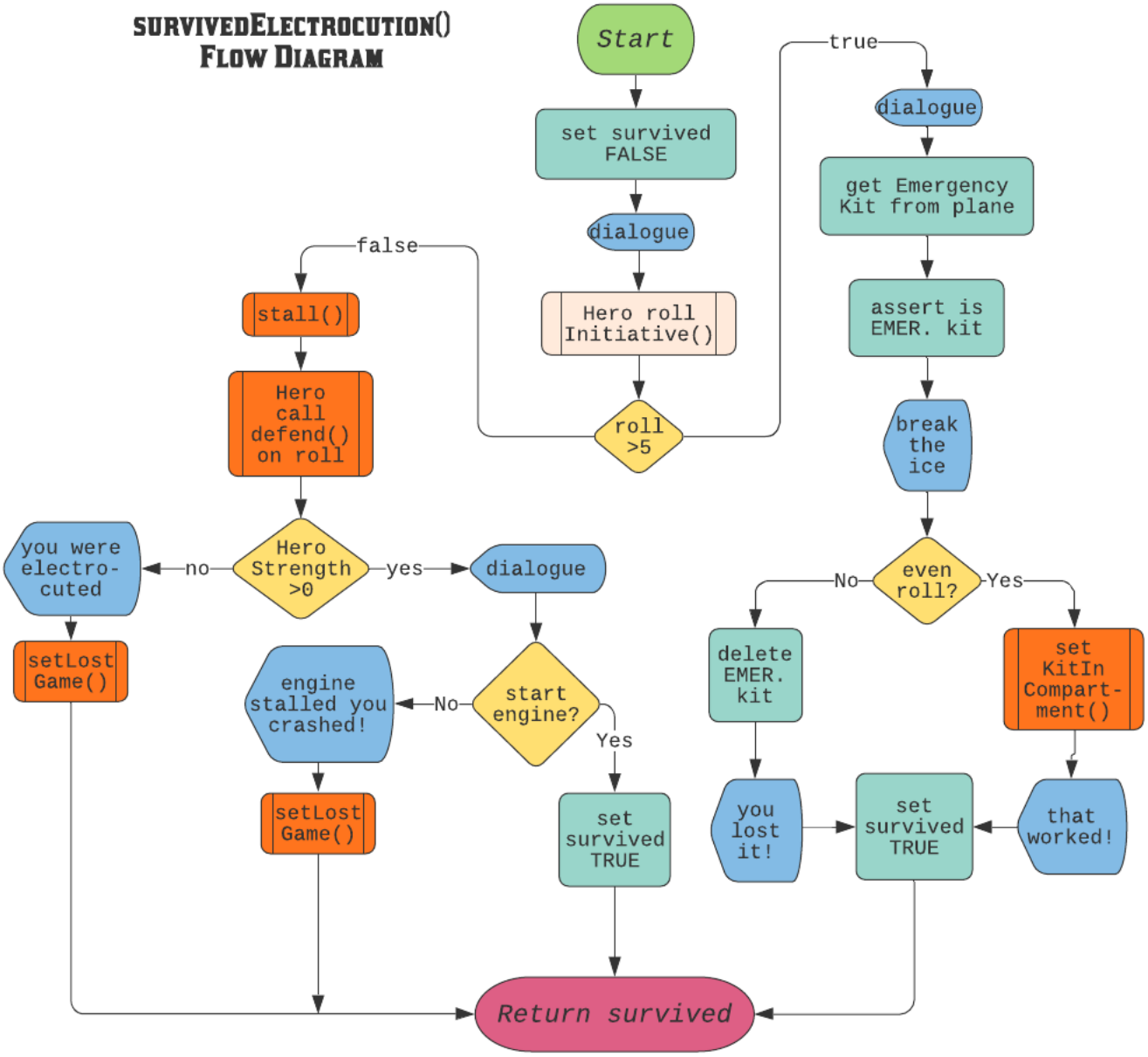
Game: A Nap in Time:
A History Major's Unexpected Journey

Lindy Level

checkPoint10() Flow Diagram



survivedElectrocution() Flow Diagram

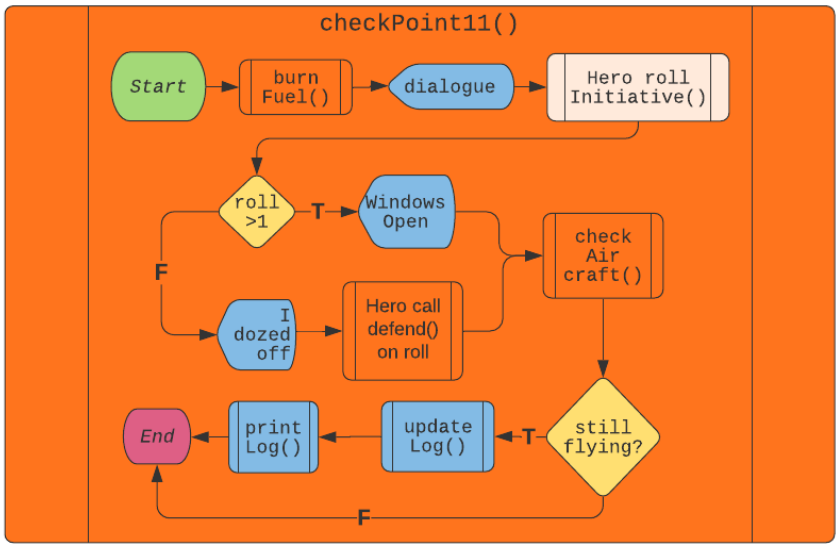


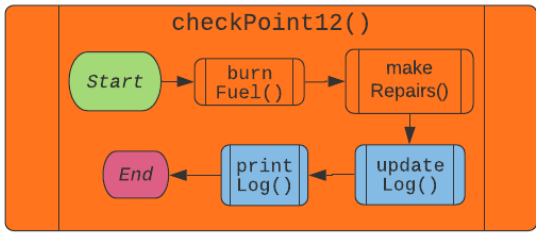
Designer: Glen Gougeon
Date: 9-20-20

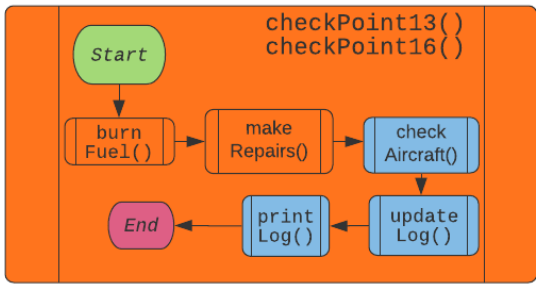
Game: A Nap in Time:
A History Major's Unexpected Journey

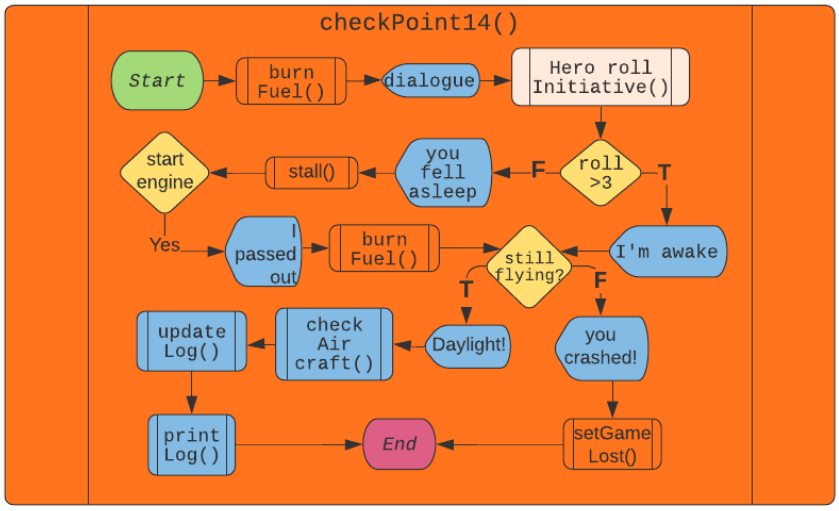
Lindy Level

checkPoint() Flow Diagrams-2







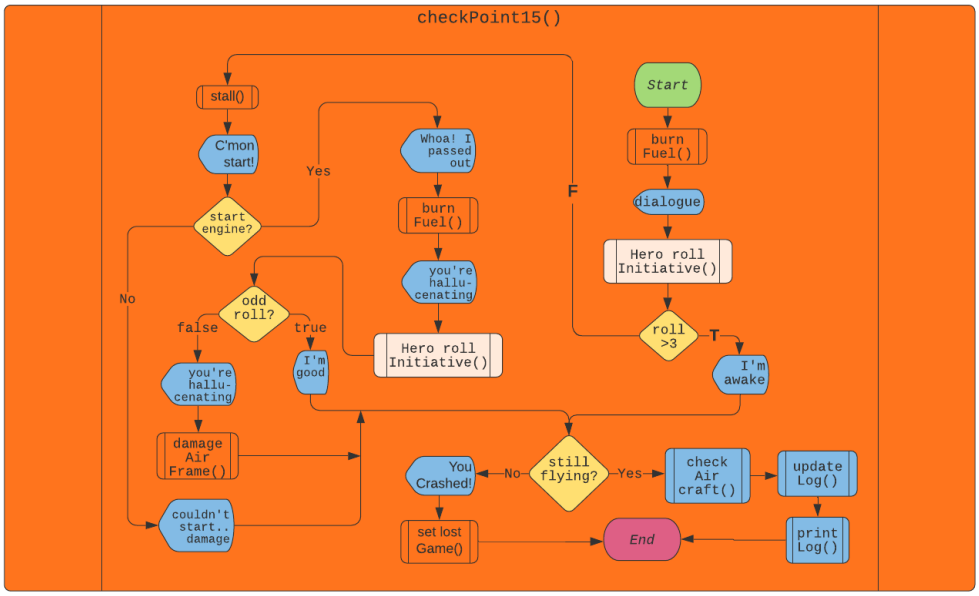


Designer: Glen Gougeon
Date: 9-20-20

Game: A Nap in Time:
A History Major's Unexpected Journey

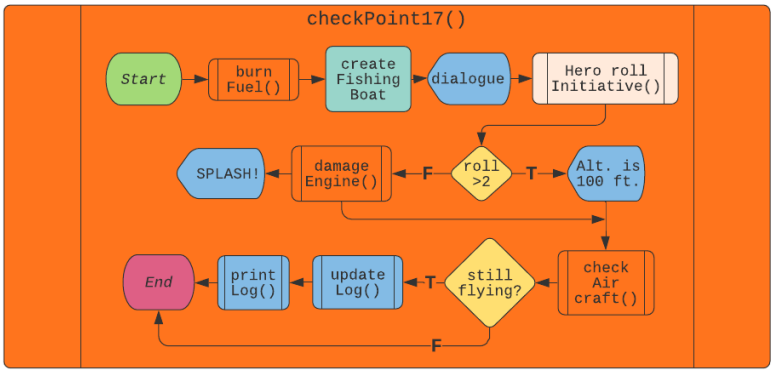
Lindy Level

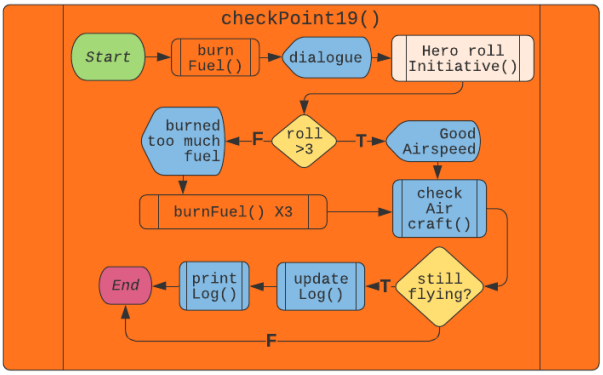
checkPoint15() Flow Diagram

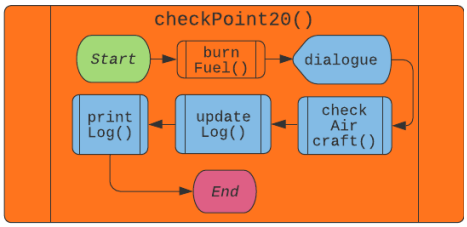


Lindy Level

checkPoint() Flow Diagrams-3

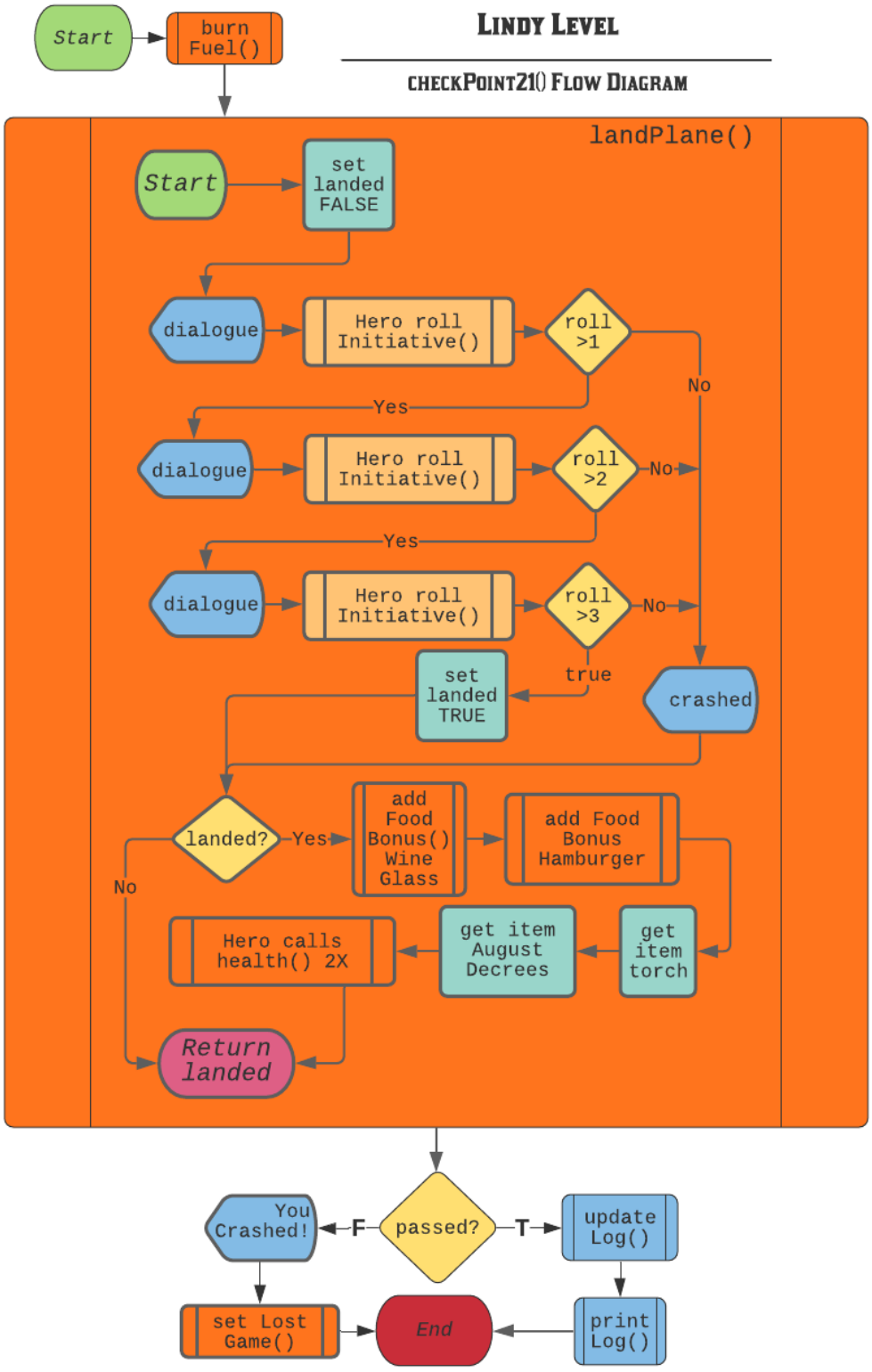






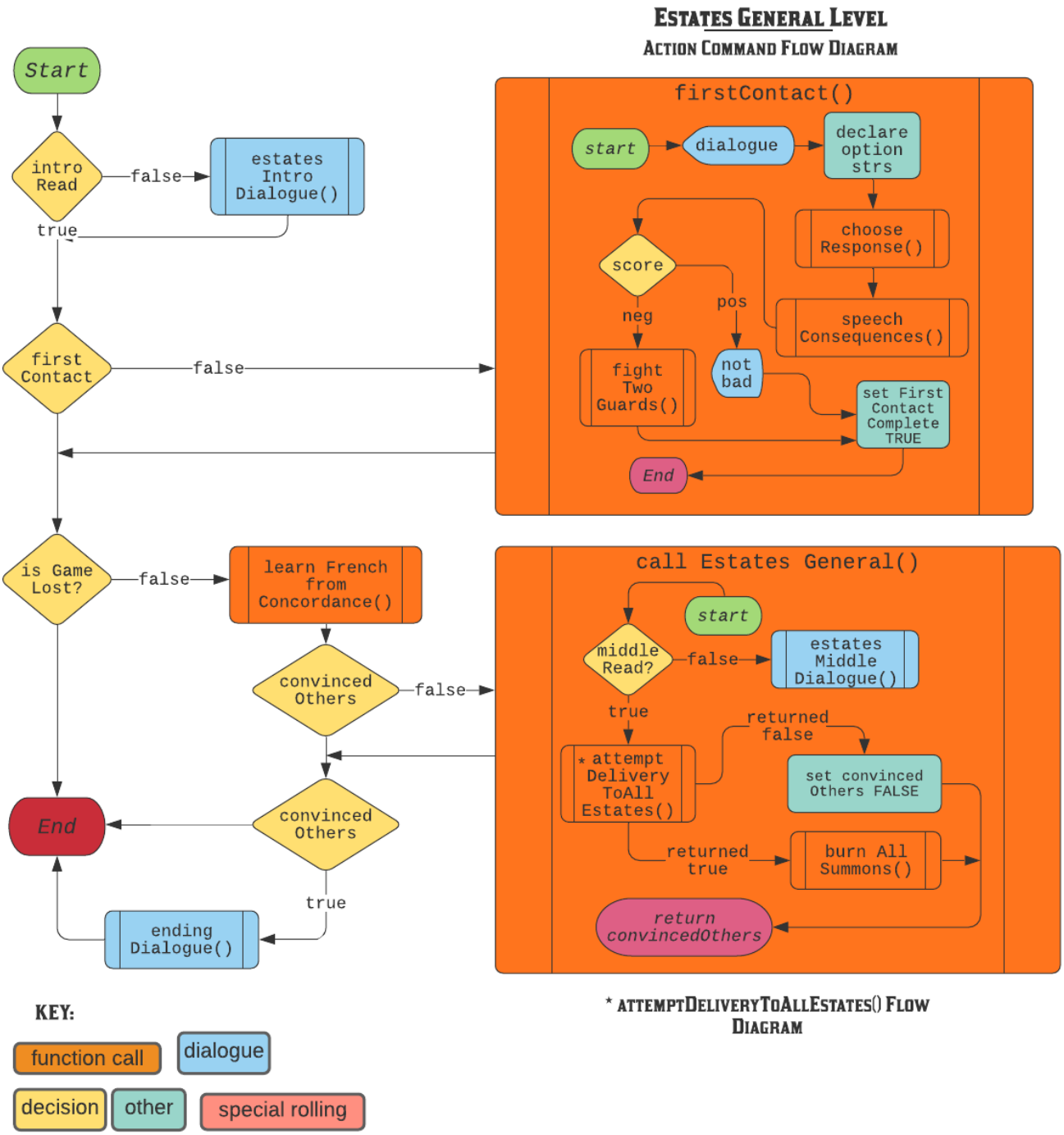
Designer: Glen Gougeon
Date: 9-20-20

Game: A Nap in Time:
A History Major's Unexpected Journey



Designer: Glen Gougeon
Date: 9-20-20

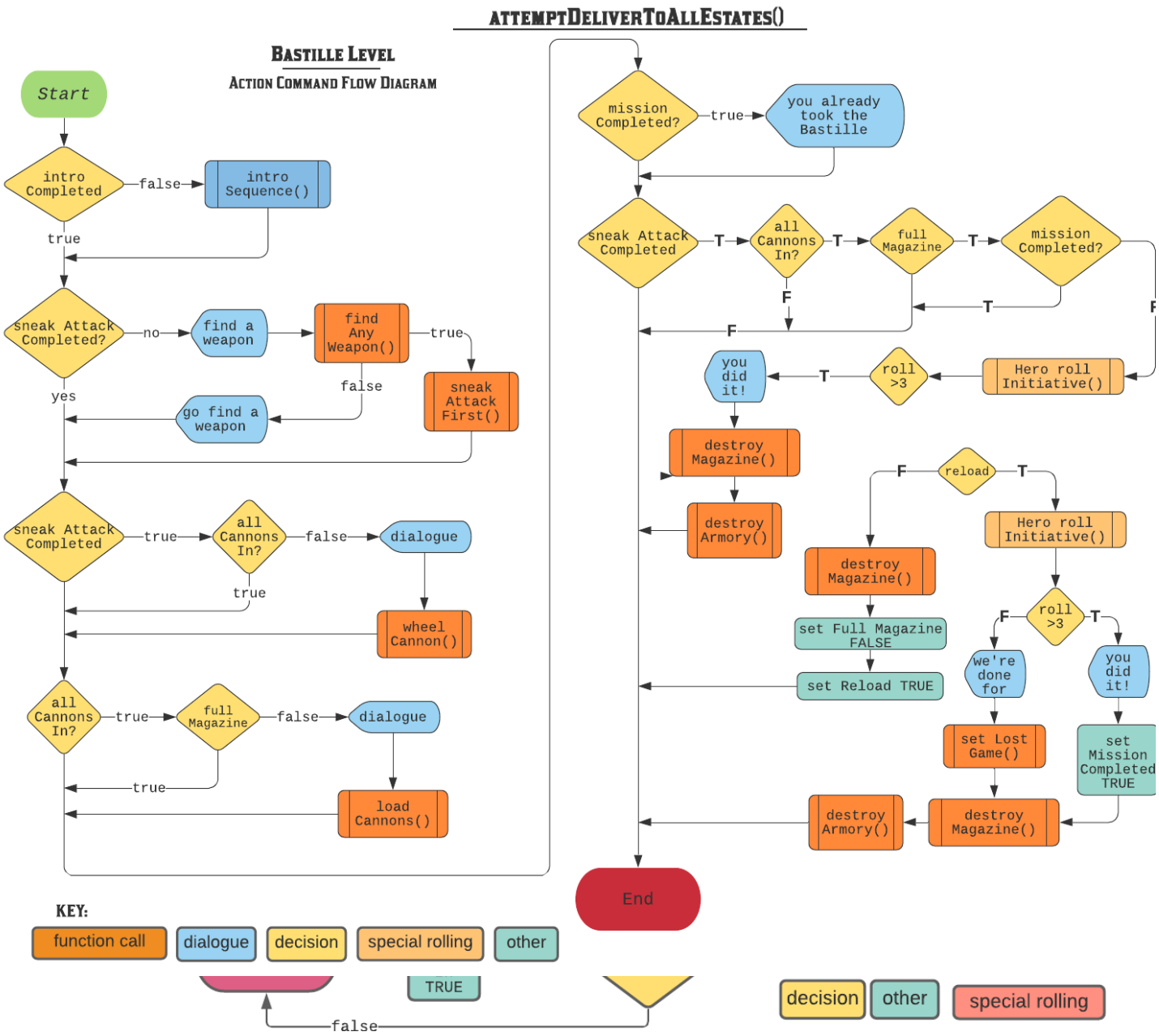
Game: A Nap in Time:
A History Major's Unexpected Journey



* ATTEMPTDELIVERYTOALLESTATES() FLOW DIAGRAM

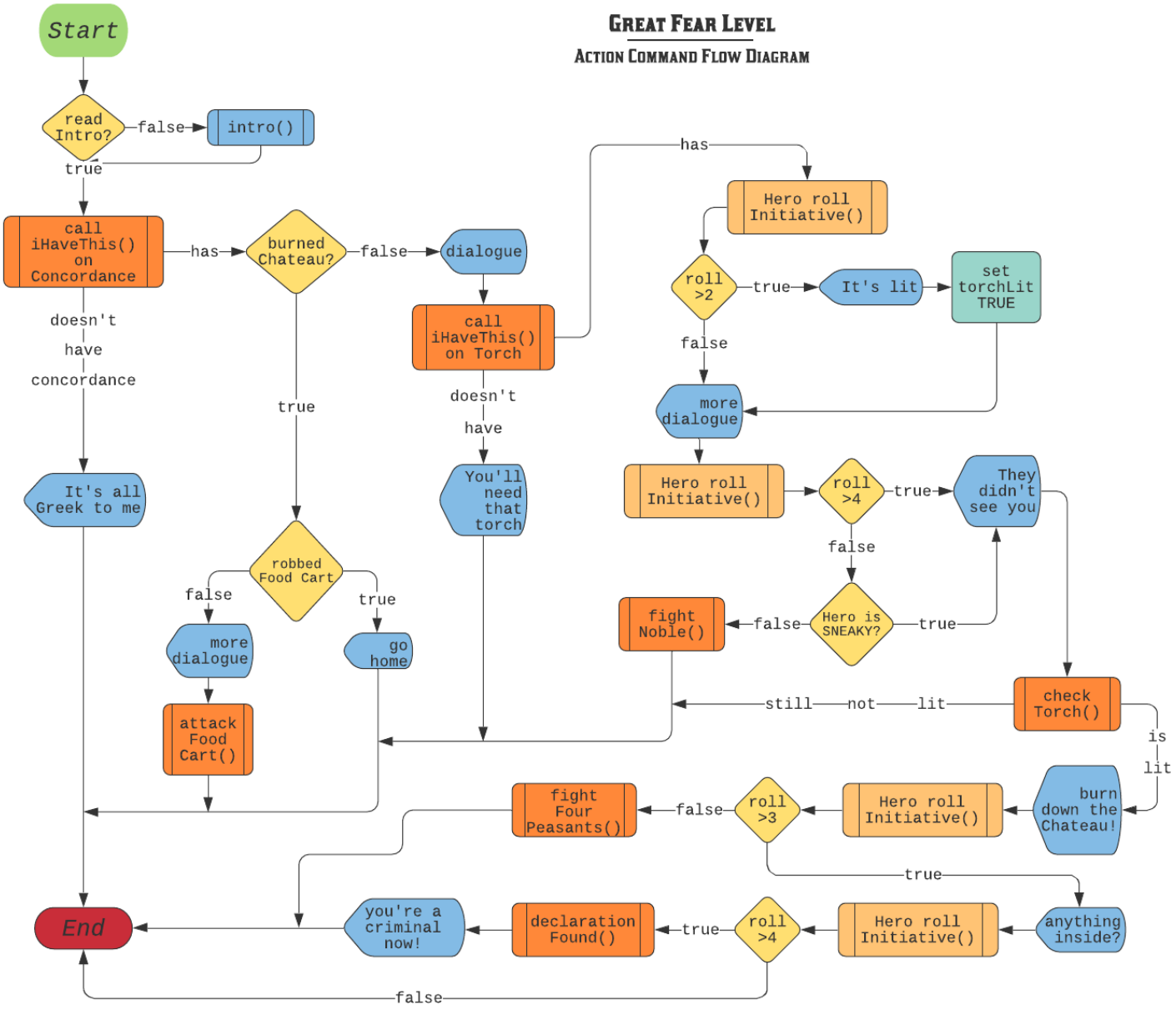
Designer: Glen Gougeon
Date: 9-20-20

Game: A Nap in Time:
A History Major's Unexpected Journey



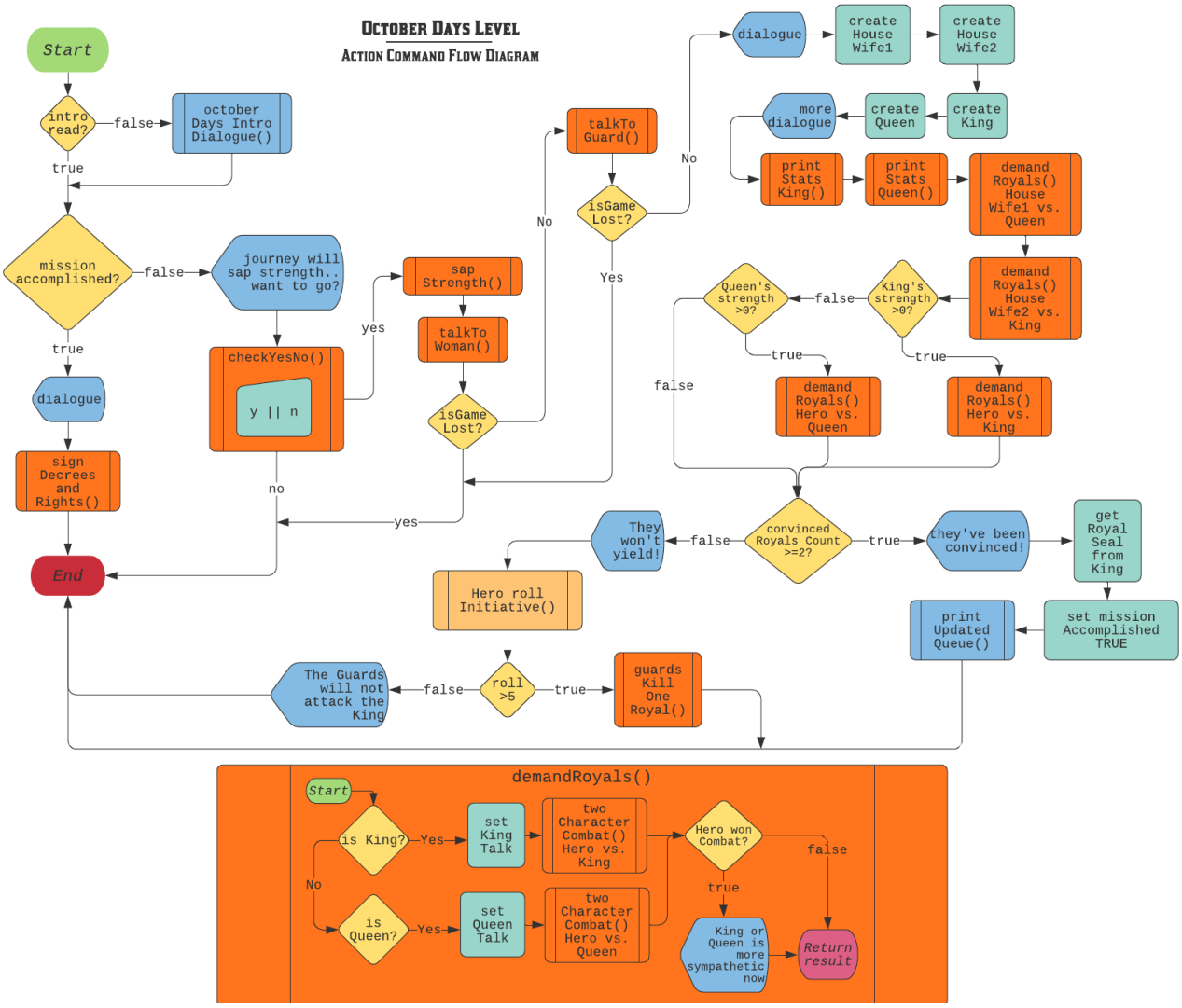
Designer: Glen Gougeon
Date: 9-20-20

Game: A Nap in Time:
A History Major's Unexpected Journey



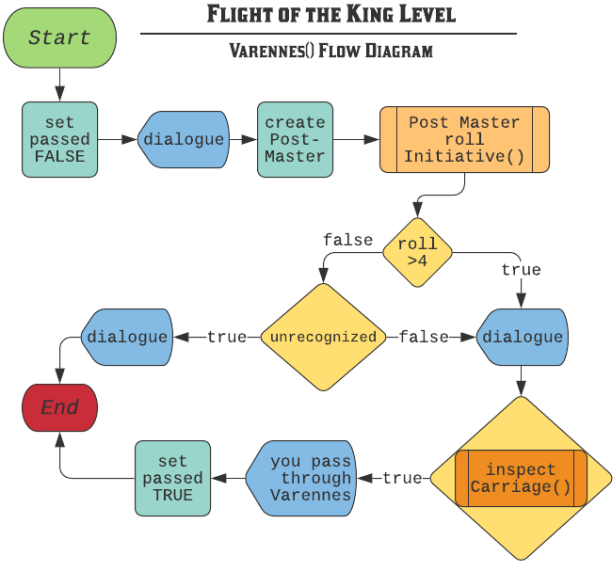
Designer: Glen Gougeon
Date: 9-20-20

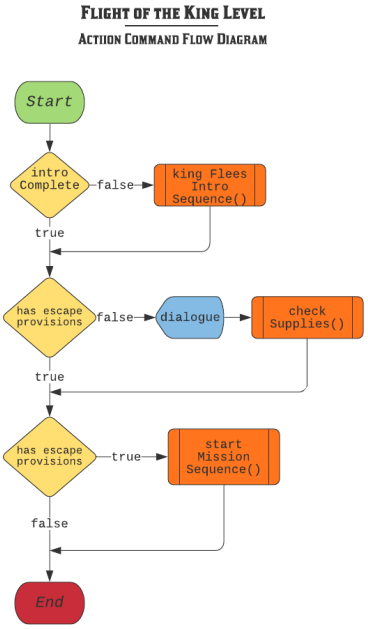
Game: A Nap in Time:
A History Major's Unexpected Journey

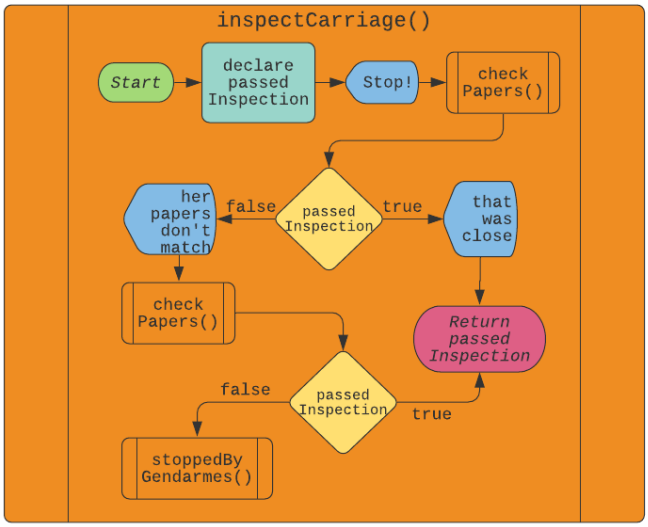


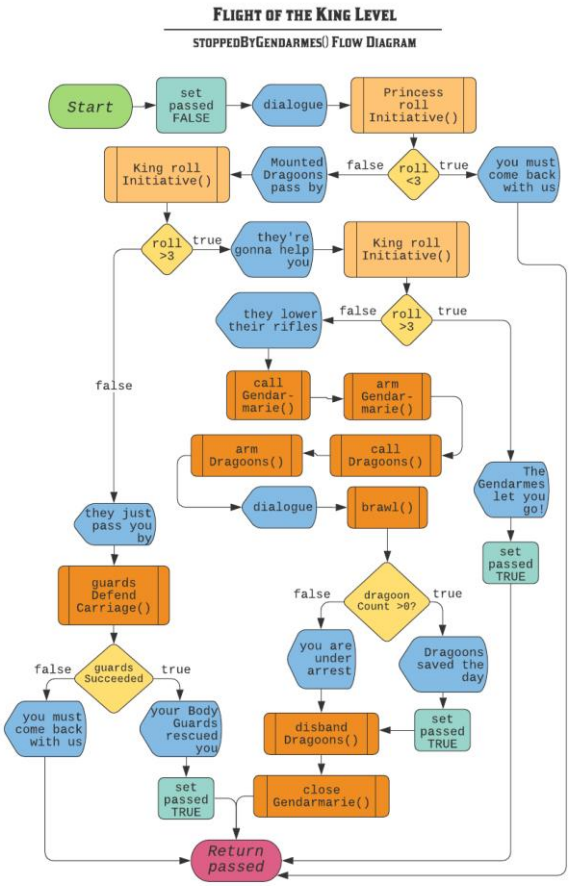
Designer: Glen Gougeon
Date: 9-20-20

Game: A Nap in Time:
A History Major's Unexpected Journey



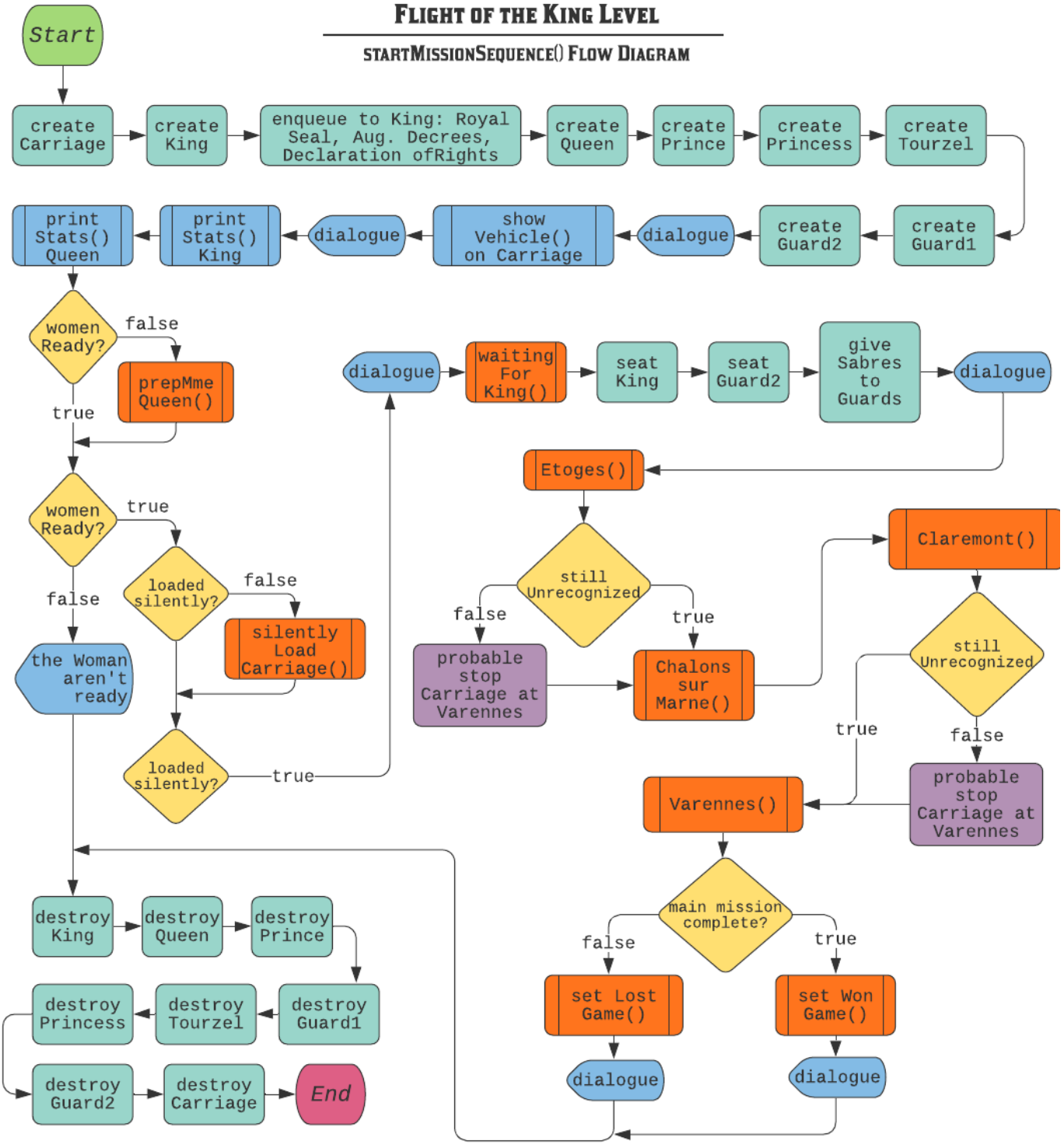






Designer: Glen Gougeon
Date: 9-20-20

Game: A Nap in Time:
A History Major's Unexpected Journey



Designer: Glen Gougeon
Date: 9-20-20

Game: A Nap in Time:
A History Major's Unexpected Journey

DATA STRUCTURES:

Stack. First-In-Last-Out.
Every level included the stack data structure for holding items. Items here refer to the Item Class that I defined for the game. Items were passed between Item pointers (Item*) Stacks and Queues extensively.

The level stack called 'itemStack' contains items pushed onto it by the level's corresponding constructor when called by the Game class. The stack defined and used a StackNode as a private member of the stack class. This was an obvious advantage of using C++ to program games, since it reduces the number of files, and extra compilation to implement a stack while at the same time ensuring that the structure is kept securely nested within the stack class.

Queue. First-In-First-Out.
Every Character contained a queue data structure named 'satchel'. The variant of queue used was a circular queue. There are no sentinels used, and every reference is to the head. The queue defined and used a QueueNode for the set of links required for the structure. Each QueueNode held an Item within it, similar to the StackNode described for the Stack Class. The queue pointer named 'droppedItems' was used to store items the user wished to leave at a level, and for the newly discovered items the user did not want.

Conclusion

A Nap in Time, was an adaptation of a game I developed for my Senior project, which involved copywritten works, and themes of which I will not mention here. Structurally, I have improved on the original structure in various small ways, but is essentially the same game, with a different theme.

I heavily refactored the primary runGameShell() function, in particular allowing the user to specifically enter missions, whilst still allowing them to leave levels to search for items etc. The game theme is highly scalable. I had intended to have levels for the entire French Revolution up to the time of Napoleon for example, but that proved too time consuming a task.

Improvements. I would incorporate the use of arrow keys with highlighted menu functionality instead of the cumbersome use of the home key row, with integers. A game map that the user could reference as they discovered levels, and the ability to navigate using a vehicle would have been great.. The intention was to show my ability to handle a complex architecture using inheritance, data structures, memory management, and some basic object-oriented design principles, which I believe I have shown, if you look at the code base, and by means of the game's reliability.