Tales of Gods and Treachery

The Hunt for Bálbrandr

Developers

Lead Programmer

Jose Uusitalo

Lead Writer

Léa Subrenat

Lead Quality Assurance & Database Management

Tuomas Kuutti

Cool Statistics

- Game Design Document: 11 days
- Programming: 23 days
- 2-3 hours of gameplay.
    - 55 minutes if you know what to do beforehand.
- 194 commands for an ideal playthrough.
- Estimated 250 hours spent programming.
- 3200 words of dialogue. (4.5 A4)
- 10 100 words of English in total. (14.5 A5)

Code Statistics

- 49 unique items.
- 40 unique characters.
- 72 recognized command aliases and 32 unique commands.
- 86 locations.
- 600 records in database.
- 2100 lines of code (no empty or comment lines)
- 37 Python files.

Content

- 2 unique endings, 6 variants of the other.
- Two separate, but intertwined storylines.
- 10 chapters.
- Some optional content.
    - Extra locations, characters, items, etc.

Setting, Story & Myths (1)

- Game world: initial brainstorming
    - Countries and cultures
    - Adventure
- Research
    - History
    - Myths
    - Languages

Setting, Story & Myths (2)

- Start of the writing process
    - Chronological order
    - Modification of the myths
    - Details (descriptions)
- Incorporation in the game
    - NPCs, items

Setting, Story & Myths (3)

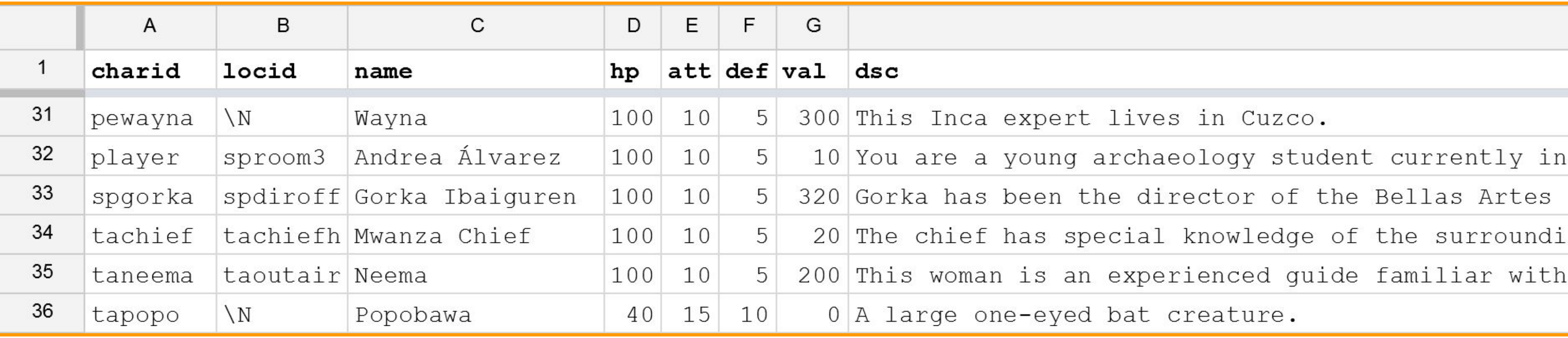
You walk to the cab and the driver opens the window. He looks at you from the front seat. His wide smile reveals a row of dark, stained teeth. "To Eleusa Monastery? Right away." He turns the key in the ignition and the car coughs loudly. "Did you know "Eleusa" comes from Byzantine Greek and means "the Tender?" It's probably not that interesting to you though, I'll concentrate on driving again." Georgi faces the road and the car jumps forward.

Gorka Ibaiguren’s office depicts his two passions clearly and leaves no room for doubt about his love of surf and art. While his desk is always well organised and professional, the walls reflect another side of his personality. The vigorous middle-aged man has attached a turquoise and yellow surfboard on the wall, right next to a large poster of Sopelana beach. On the other side of his bureau resides an even more substantial world map, with small photographs of artefacts pinned upon the locations they are from.

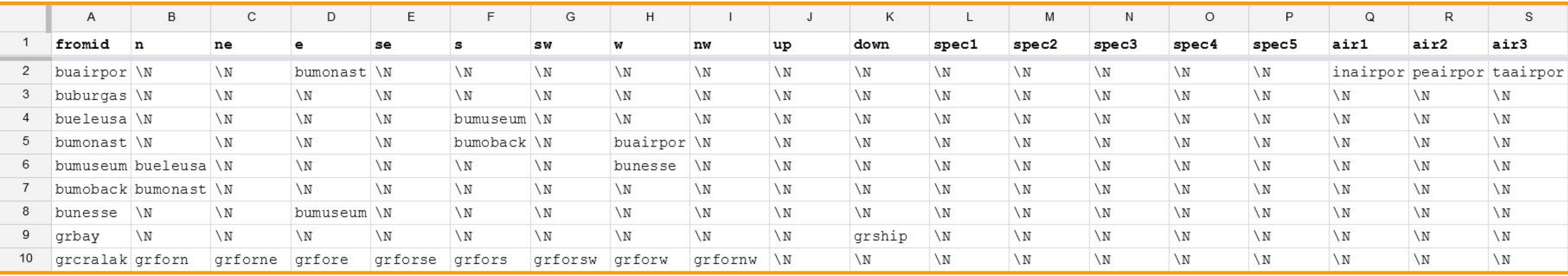
Google Drive

- Programming at the same time.
- Managing the database: days of saved time.
- Google Sheets > TSV files > MySQL

Google Sheets Database



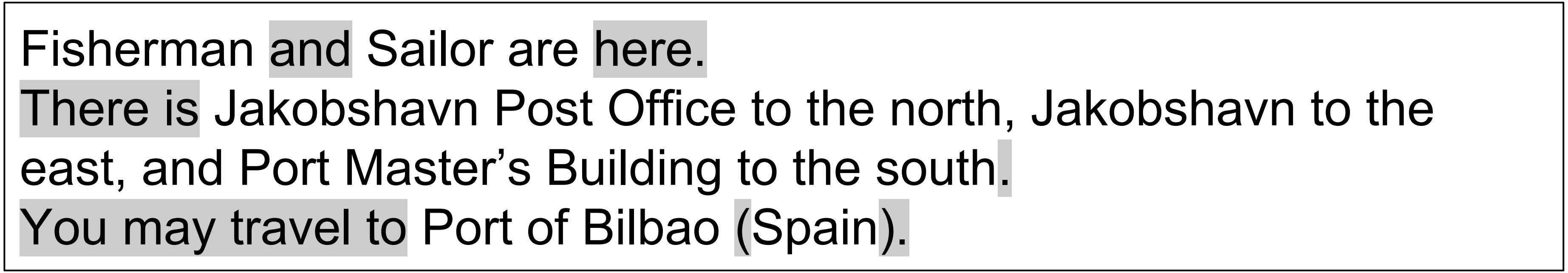
	A	B	C	D	E	F	G	
1	**charid**	**locid**	**name**	**hp**	**att**	**def**	**val**	**dsc**
31	pewayna	\N	Wayna	100	10	5	300	This Inca expert lives in Cuzco.
32	player	sproom3	Andrea Álvarez	100	10	5	10	You are a young archaeology student currently in
33	spgorka	spdiroff	Gorka Ibaiguren	100	10	5	320	Gorka has been the director of the Bellas Artes
34	tachief	tachiefh	Mwanza Chief	100	10	5	20	The chief has special knowledge of the surroundi
35	taneema	taoutair	Neema	100	10	5	200	This woman is an experienced guide familiar with
36	tapopo	\N	Popobawa	40	15	10	0	A large one-eyed bat creature.



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
1	**fromid**	**n**	**ne**	**e**	**se**	**s**	**sw**	**w**	**nw**	**up**	**down**	**spec1**	**spec2**	**spec3**	**spec4**	**spec5**	**air1**	**air2**	**air3**
2	buairpor	\N	\N	bumonast	\N	\N	\N	\N	\N	\N	\N	\N	\N	\N	\N	\N	inairpor	peairpor	taairpor
3	buburgas	\N	\N	\N	\N	\N	\N	\N	\N	\N	\N	\N	\N	\N	\N	\N	\N	\N	\N
4	bueleusa	\N	\N	\N	\N	bumuseum	\N	\N	\N	\N	\N	\N	\N	\N	\N	\N	\N	\N	\N
5	bumonast	\N	\N	\N	\N	bumoback	\N	buairpor	\N	\N	\N	\N	\N	\N	\N	\N	\N	\N	\N
6	bumuseum	bueleusa	\N	\N	\N	\N	\N	bunesse	\N	\N	\N	\N	\N	\N	\N	\N	\N	\N	\N
7	bumoback	bumonast	\N	\N	\N	\N	\N	\N	\N	\N	\N	\N	\N	\N	\N	\N	\N	\N	\N
8	bunesse	\N	\N	bumuseum	\N	\N	\N	\N	\N	\N	\N	\N	\N	\N	\N	\N	\N	\N	\N
9	grbay	\N	\N	\N	\N	\N	\N	\N	\N	\N	grship	\N	\N	\N	\N	\N	\N	\N	\N
10	grcralak	grforn	grforne	grfore	grforse	grfors	grforsw	grforw	grfornw	\N	\N	\N	\N	\N	\N	\N	\N	\N	\N

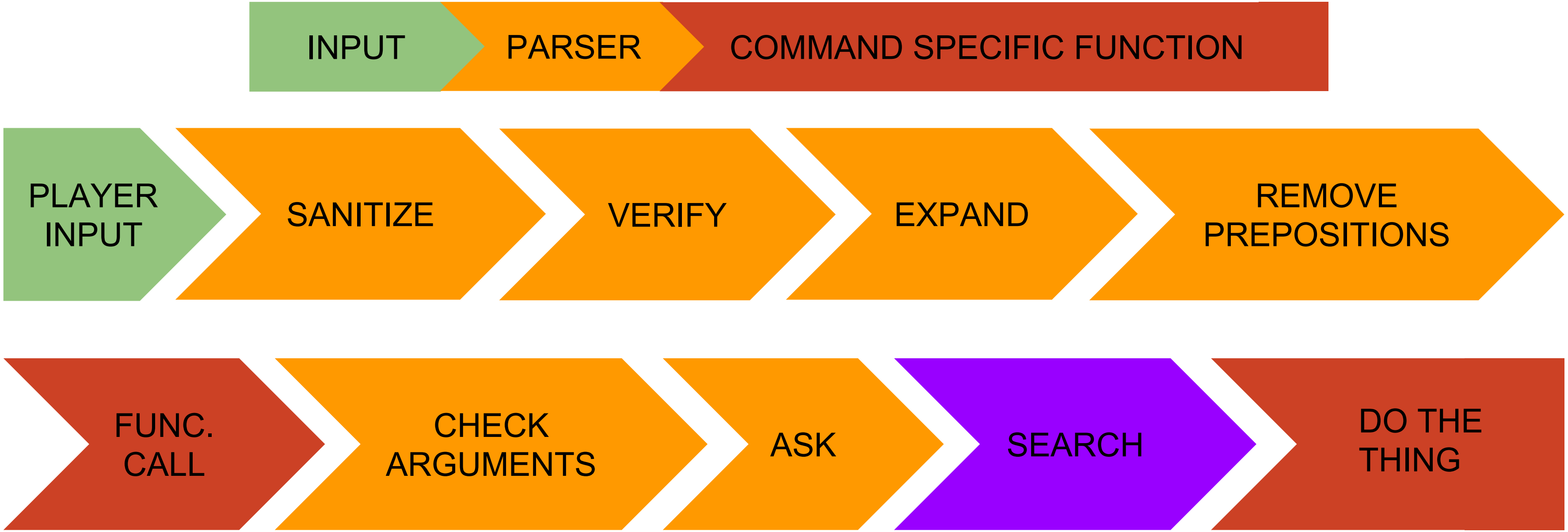
Modularity

- 98 functions.
- Generic, very little hardcoding.
- Swap the database: new game.
    - Almost…



Fisherman and Sailor are here.
There is Jakobshavn Post Office to the north, Jakobshavn to the east, and Port Master’s Building to the south.
You may travel to Port of Bilbao (Spain).

Parser Structure



INPUT → PARSER → COMMAND SPECIFIC FUNCTION

PLAYER INPUT → SANITIZE → VERIFY → EXPAND → REMOVE PREPOSITIONS

FUNC. CALL → CHECK ARGUMENTS → ASK → SEARCH → DO THE THING

Command Walkthrough (0/8)

```
G BULG D TO GORKA
```

Command Walkthrough (0/8)

G BULG D TO GORKA

Command Walkthrough (1/8)

- Remove “bad” characters like: % ; & _ = $,
  - First line of defence against injections.
  - Convert to lowercase.
- Split input into a list by space characters.
- Remove empty items: remove whitespace.

Input: [“g”, ”bulg”, ”d”, ”to”, ”gorka”]

Command Walkthrough (2/8)

- Read first word, verify command validity.
- Read into memory, delete from input list.
- Expand command: match alias to command.
    - g -> give
- Strip prepositions.

Input: [”bulg”, ”d”, ”to”, ”gorka”]

Command Walkthrough (3/8)

- Real command in memory.
- Player input in a single list.
- Good to go!
- Execute function with the same name as the command with the input list as parameter.

Execute: give ([”bulg”, ”d”, ”to”, ”gorka”])

Command Walkthrough (4/8)

- Back to parser!
- Check for words to delete at start of string.
  - For example delete “to” in: TRAVEL TO AIRPORT
- Check for required words.
  - “to” in give, “on” or “with” in use, “about” in talk
- Check if player input anything at all:
  - If not, ask for input

Command Walkthrough (4/8)

- Index position of required word!
  - In our example, the index would be 2.
- Return input back to the give function along with the index.

Return: [”bulg”, ”d”, ”to”, ”gorka”] , 2

Command Walkthrough (5/8)

- Take the index 2 and split the input list in two.

Input: [”bulg”, ”d”, ”to”, ”gorka”]

Inputs: [”bulg”, ”d”] and [”gorka”]

- First list: Item in player inventory
- Second list: Character in the same area

Command Walkthrough (6/8)

- What is [”bulg”, ”d”] ?
- Search with wildcards: *bulg*d*
  - One potential match: “a bulging golden wallet”
- A necessity when you have items with such names as Guðhjǫrr.
- Problem: Found two items.
  - Bulgarian Dagger (ID: budagger)
  - Bulgarian Dictionary (ID: dicbulg).

Command Walkthrough (7/8)

- Ask player!
- Keep asking until only one item is found.
- Return the ID: budagger
- Do the same thing with the other input list, just search for characters.

Inputs: [”budagger”] and [”spgorka”]

Command Walkthrough (8/8)

- What we now know.
  - Do what: give
  - Give what: budagger
  - Give to: spgorka
  - Give from: player

G BULG D TO GORKA

```
transfer ( [“budagger”] , [“spgorka”] , [“player”] )
action.give ( [“budagger”][0] , [“spgorka”][0] )
```

Actual Debug Messages

```
> g bulg d to gorka
[DBG] parse STRIPPED INPUT: give ['bulg', 'd', 'to', 'gorka']
[DBG] checkArgs: (['bulg', 'd', 'to', 'gorka'],) Give what to who? to
[DBG] checkArgs RETURNS: ['bulg', 'd', 'to', 'gorka']
[DBG] idFromName ARGUMENTS: ['bulg', 'd'] item ['player'] True
[DBG] idFromName INPUTLIST: ['bulg', 'd']
[DBG] idFromName INPUTWORDS POPPED: ['bulg']
[DBG] idFromName INPUTNAME CLEANED: bulg
[DBG] idFromName INPUTLIST: ['d']
[DBG] idFromName INPUTWORDS POPPED: ['bulg', 'd']
[DBG] idFromName INPUTNAME CLEANED: bulg%d
[DBG] idFromName RETURNS: ['budagger', 'dicbulg']
[DBG] ask.which: ['budagger', 'dicbulg'] item ['player']
[DBG] ask.which NAMES ['Bulgarian Dagger', 'Bulgarian Dictionary']
Did you mean Bulgarian Dagger or Bulgarian Dictionary? (empty to
cancel)
> gg
[DBG] idFromName ARGUMENTS: ['gg'] item ['player'] True
[DBG] idFromName INPUTLIST: ['gg']
[DBG] idFromName INPUTWORDS POPPED: ['gg']
[DBG] idFromName INPUTNAME CLEANED: gg
[DBG] idFromName RETURNS: ['budagger']
[DBG] ask.which PLINPUT: ['budagger']
[DBG] ask.which RETURNS: ['budagger']
[DBG] idFromName ARGUMENTS: ['gorka'] people [] True
[DBG] idFromName INPUTLIST: ['gorka']
[DBG] idFromName INPUTWORDS POPPED: ['gorka']
[DBG] idFromName INPUTNAME CLEANED: gorka
[DBG] idFromName RETURNS: ['spgorka']
[DBG] invSlotByItemID SLOT: None ['spgorka']
[DBG] invSlotByItemID SLOT: ['budagger'] ['player']
Bulgarian Dagger given to Gorka Ibaiguren.
[DBG] action.give: budagger spgorka
"You are supposed to inspect this artifact. Not me."
[DBG] invSlotByItemID SLOT: None ['player']
[DBG] invSlotByItemID SLOT: ['budagger'] ['spgorka']
Bulgarian Dagger given to Andrea Álvarez.
```

Tales of Gods and Treachery

The Hunt for Bálbrandr

Questions?