

EXPLORATION OF MEDIEVAL EUROPE THROUGH INTERACTIVE FICTION

By

Kyle Willey

APPROVED:

Karen Bruhn Director

Joseph Foy Second Committee Member

Rebecca Viles Third Committee Member

ACCEPTED:

Dean, Barrett, the Honors College

Interaction is key to education, as students who perform their own inquiry into a subject retain information longer. The field of interactive fiction, which emphasizes personal decision making and freedom of choice, is ripe for opportunity as it is relatively simple to develop and deploy to audiences of any size. However, few interactive fiction platforms exist with the openness and flexibility required for classroom use. My project attempted to create an interactive fiction platform that can be created for and engaged with by both teachers and students. This led to the creation of an interactive fiction platform that conforms to a variety of requirements, such as openness and compatibility across multiple platforms, and which can display meaningful content. This was accomplished by someone with a content area education background and only limited computer science experience, and shows promise for similar future endeavors.

Contents

Chapter 1. Historical Research on Medieval Poland.....	2
Integrating Real Sites for Authenticity	6
Conflicting Research and Quality of Sources	7
Chapter 2. Motivations for Interactive Fiction.....	9
Chapter 3. Writing Process for a Historical Fiction Narrative.....	13
Writing in Acts	17
Integrated Review	19
Chapter 4. Integration of Game Elements with Narrative Structures.....	20
Chapter 5. Technical Aspects of Interactive Fiction Deployment.....	22
Human-Content Interface Philosophy	22
XMICAE for the Layperson	23
A Computer Scientist's Approach to XMICAE	24
Chapter 6. Reflections and Future Applications.....	35
Illustrations.....	37
Historical Glossary.....	39
Technical Glossary.....	40
Works Cited.....	41

Chapter 1. Historical Research on Medieval Poland

My research is divided in sources by topic of study—sources on Polish history, medieval life, and an emphasis on the Fourth Crusade were all used independently of each other, as none of the available sources had all of these elements available in a combined package. The University of Buffalo maintains an excellent portal for scholarly documents about medieval Poland, and it turned me towards the Halecki books that I used as a central canon for my research on Poland. Information for the other parts of my research comes from Polish-focused history books, geographical and ecological studies of Poland, and a variety of secondary sources. Much of the information about the daily life of a medieval Pole is somewhat difficult to deduce based on records created by or of the Polish people themselves. Polish was not used as a written language until well after the early 1200's, and there does not seem to have been a focus on record-keeping akin to what was seen in England with the Domesday Book. Much of my research toward medieval Polish lifestyles is extrapolated from my own experiences with Slavic culture and the canon of research that has been undertaken on the average medieval person, with some effort being made to respectfully portray both the uniqueness and similarities of the Polish people in the Middle Ages.

Much of my research focused on deducing the political climate of Poland, which could charitably be described as volatile. Despite the relatively small role that the politics of Poland played in the narrative directly, it was a crucial part of understanding how crusaders and German influences would be portrayed as the player encountered them. Mieszko III the Old—the immediate predecessor of Leszek I Biały, who is the High Duke of Poland in 1203—ascended the throne of Krakow a total of four times, and was the last High Duke who still could be said to

truly control the province immediately surrounding Krakow with the same claim as he had to the throne; Leszek Bialy would attempt to reclaim some of this prestige, but ultimately would not be successful in reasserting the relationship between his title of High Duke and claims to the lands surrounding Krakow (Gieysztor *History of Poland* 75; 2nd ed.). Much of my focus on the life of the medieval Pole comes from studies of Western Europeans, and attempting to determine what elements of modern Polish life existed back in the medieval times. Particularly helpful assets in my research were the works of Frances and Joseph Gies, particularly their series of books on life in medieval castles, cities, and villages. Nothing in my research seems to suggest that the use of sources that predominantly address other parts of Western Europe is a source of great inaccuracy; indeed, medieval Poland was heavily influenced by German expansion and other European immigrants (Gieysztor *History of Poland* 97; 1st ed.) as well as trade with neighbors from both Eastern and Western Europe (Gieysztor *History of Poland* 75; 2nd ed.). while the Polish political system may be slightly different from that of what would typically be considered Western Europe, the country is Roman Catholic and has followed that faith since Mieszko I adopted Christianity in 966. Though Gothic style has reached the country by 1203, it has done so only recently, and there is still a strong Romanesque influence in culture and architecture (Davies *God's Playground: a History of Poland* 65). As such, Poland has much in common with the Germans to the west—in part owing to the fact that German settlements inside Poland were common, particularly in the westmost portions of Poland—while it has very little of its social order in common with its Eastern Orthodox neighbors to the East. However, I also attempt to integrate specific elements of Slavic culture into the work when possible, especially when the player reaches Torun, which has not yet become a major city despite being on the

major trade route of the Vistula ("Medieval Town of Toruń.").

Another misconception about Poland is that it was a “minor” player in the medieval times. This largely seems to be a product of German historians who were very negative toward the country's contributions to medieval society, no credible research from an unbiased source seems to suggest that Poland lagged behind the rest of Europe in terms of technological or cultural advances. In fact, many of Poland's large cities were trading hubs and the country's perceived lack of participation in international affairs seem to be due to the region's instability or due to Poland's miserably poor presence in the modern field of history, which has not been helped by the events of the twentieth century. Much of the sources for Polish history are not available to a historian who is only familiar with English and Latin, as many of the best histories of Poland exist only in Polish, though a solid overview is available at <http://info-poland.buffalo.edu/web/history/early/link.shtml>, which served as the portal for much of my early research, and provides links to many translated Polish histories, including Halecki's work, which quickly became a cornerstone of my research.

The exact time-frame for my narrative is left somewhat vague in the text, but it is envisioned to begin around June or July of 1203. This lets us remain somewhat historically accurate; the question could be raised about why the protagonist took over six months after the start of the siege of Zara in November of 1202 to actually leave and go home, and some more questions could be asked about the way that the protagonist would go about doing so, but these are matters of interest and do not necessarily preclude the scenarios that unfold; I leave them unanswered because they do not really need to be answered. I chose the summer of 1203 so that the protagonist can still operate in the political culture that spawned the crusade, as well as to

give a more friendly climate for travel. Passing through Poland in winter would be much more difficult than it would be in the summer or fall. Another upside of this is that it allows the player to begin to see some of the unintentional side effects that have begun to occur as a result of foreign colonization in Poland; the player encounters members of the Knights of the Sword, who Halecki briefly details as a knightly order with ties to the crusaders (*Borderlands of Western Civilization: a History of East Central Europe* 81-82). Time passes in a somewhat nondescript fashion throughout the course of events; some of the endings depend on how long the player takes, though based on estimates of medieval travel this takes around a month, allowing the player's character to return to Western Europe in August if the player is particularly effective, Of course, this could take a significantly longer time if the player gets their character critically injured or otherwise held up during the course of events.

Another element of the choice of time comes with the current High Duke of Poland. Leszek the White is responsible for a number of important changes in Poland that ultimately lead to a massive loss in power. In 1203 he has just come to power in 1202, That the narrative takes place almost exclusively within important holdings of Spindleshanks is largely coincidental; the protagonist enters from the south-eastern border, in the lands surrounding Krakow, continues to Krakow, then winds up traveling north to Torun and Greater Poland, which Spindleshanks had a claim on even before he would go on to become High Duke of Poland following Leszek (Gieysztor *History of Poland* 81; 2nd ed.). It is worth noting that Spindleshanks may have actually been more powerful than Leszek; although he is the latter's vassal he has exerted claims over the provinces of Greater Poland, Lesser Poland, and Krakow—while it is not explicitly stated, influence of the court may have been behind Leszek I's decision to split off Krakow as a

province from the title of High Duke of Poland (Gieysztor *History of Poland* 90; 1st ed.).

Though it seems fairly certain that Leszek's rule was troubled by internal factors, including the immense power that Spindleshanks has amassed, this is largely reflected in dealings outside the public sphere which the player would not be party to. Part of Spindleshanks' power comes from his ties to powerful nobles that tend to disapprove of Leszek, something that could come as a result of his choice to open up Poland even further for colonization and settlement.

Integrating Real Sites for Authenticity

The two “real” sites prominently featured throughout the course of events are Krakow and Torun. Krakow had been the central seat of power in Poland since the 11th century, and it is already a flourishing center of trade and medieval power when the player visits it. Although Krakow was later razed by the Mongols in 1241, it is believed to have been rebuilt in a mostly identical format, which actually leaves a very good picture to extrapolate what the city would have looked like in 1203 since the plans can be viewed and referenced (“History of Krakow”). The Church of St. Adalbert is one of the locations opened to the player as they explore Krakow, as it is one of the landmarks that still exists in Krakow and can serve as an example of the Romanesque architecture of the city.

Krakow, as a city, is well enough known for a conjectural map to be created based mainly on the presence of known landmarks—the Wawel area and St. Adalbert's Church being the most prominent of these—and a map is available in the game's codex to show players an estimation of what Krakow would look like on a map in 1203.

Torun, on the other hand, has less information existing for it; its presence in the game is

based off of the fact that there was a town at that location, though it exists in the narrative primarily to highlight the conflict between Poland and crusading forces and, due to the paucity of information available about it, does not have a physical representation in the codex, though some mention is made of its origin as a Slavic town prior to the influx of Germanic influences with the crusaders.

Conflicting Research and Quality of Sources

Throughout this project I have encountered difficulties with both consistency and quality of available sources; one credible source may contradict another, and as a result I frequently found myself being forced to make personal value judgments on the worthiness of the sources I encounter. Much of what I discovered in my early research has been excised from my notes or replaced with information from another, more accurate source, I typically consider the Halecki sources to be the most authoritative, followed by the Gieysztor sources, then the Davies source and the Gies sources; the Gies are unique in not writing about Poland, but rather on England and France—the main reason for their sources being of high value comes in their focus on a time period that is either during or directly after the Fourth Crusade, and the detail to which they detail medieval life. Another problem that has confronted me comes in the relatively small pool of historians from whom my research is drawn; interestingly, however, both Halecki and Gieysztor published consistent but highly altered forms of their histories in the revised editions of their work; the 1979 and 1968 editions of Gieysztor's *History of Poland* are radically different in their structure and emphasized information, while Halecki's works that I have sampled tend to touch on a variety of different approaches to the study of Poland and acquire a different perspective in that way.

I use a couple of websites run by the City of Krakow and the UN in my project; these are used mostly for cursory information, such as the physical location of sites that I examine which have already been made known to exist in other ways. For Krakow, I used the listed “History of Krakow”, as well as Google Maps; the Krakow Old Town has been preserved in form, if not function, for the last eight hundred years, and the St. Adalbert's Church and Wawel were the two main concerns of mine in defining the city. Note that the City of Krakow website states that it is the 1253 deed that centrally locates the marketplace, but other sources that talk about Krakow's destruction at the hands of the Huns in 1243 observe that the city was rebuilt practically identically as it had been before, and that the 1253 deed essentially formalized the matter without requiring any demolition of structures. The two buildings in Krakow I used Google Maps to place are St. Adalbert's Church and Wawel, both of which existed in their modern positions before 1253; the layout of the districts of Krakow featured in my project come from the work of the Gies and is conceptually rebuilt based on known facts that applied to most medieval cities, rather than an original plan.

Chapter 2. Motivations for Interactive Fiction

My creative project is an interactive fiction experience that places the user in the shoes of a crusader returning early from the Fourth Crusade, following the Siege of Zara. It places a high priority on historical accuracy, augmenting the traditional interactive fiction narrative with additional “trivia” sections that give additional facts or background information. Delivered through a web-based platform, it is capable of being experienced on every device compliant with recent HTML standards, and focuses on the protagonist's return to their home through Poland, and the various political structures and lives they interact with on their journey.

Interactive fiction and historical narratives are not a novel concept. However, many of the historically focused IF works are prone to inaccuracy or otherwise fail to convey much of the magnitude and depth of the events they recount. Many of the best historical IF experiences focus on already well-documented and explored historical sites and events, and far too many fall victim to over-interpretation and other conflicts with reality. My creative project, on the other hand, attempts to present a slice-of-life view of the medieval world through the eyes of a traveler. Many of the interactive fiction narratives that exist that focus on medieval times are focused either on character drama or on a heroic narrative; my goal is to essentially strip away many of the narrative stylings and focus instead on telling a story that retains some entertainment value but also draws the reader through a variety of environments and situations in which they glean knowledge about the daily life of the medieval Polish citizen. Unfortunately, as Norman Davies points out, “Cultural life in Piast Poland is largely obscured... secular society was overwhelmingly illiterate... such artifacts and writings that have survived are mainly the products of imported Catholic religious influences.” (Davies *God's Playground: a History of Poland* 65).

I feel that I have a responsibility to disclose my motivations and background for this project to give some context for the reader. First, I have always been interested in medieval history; my choice of the crusades and Poland as the setting stem from the interesting status that both this time and place have in medieval history. The crusades happened in several forms, with each occurring within unique political and geographic circumstances. I find that the Fourth Crusade has one of the most interesting interactions between the Catholic Church and secular power, particularly with regards to the role of the crusader. This is especially apparent when one gets into some of the events that occur later in the crusade and culminate in the formation of the Latin Empire.

Poland in particular makes up a good part of both my research and the setting for my project. The region is interesting in part because it exists largely on the fringe of medieval studies. Some of this relates to more modern political and historical developments that I will talk about later, but much of the lack of information on medieval Poland has to do with the fact that Poland, and the Piast dynasty which ruled it at the time of the Fourth Crusade, was fairly unstable. Falling outside the boundaries of the Holy Roman Empire Poland never wielded a tremendous amount of political or military power during the Middle Ages, and as such it is often overlooked. Poland's political climate is also remarkably tumultuous around the time of the Fourth Crusade, with some disagreement in historical circles about who is actually in charge. I find that it is likely that Leszek I Biały was High Duke of Poland, though this will be discussed more later in this commentary. Aleksander Gieysztor's sources indicate that a deal of confusion over the matter comes from the fact that Leszek or his immediate predecessor separated the ownership of certain lands that had typically been associated with the High Duke from that title,

and made them be subject to their own succession and ownership; this does not seem to have moved the seat of Poland from Wawel in Krakow—if it did, no mention was made of this—but it certainly meant that the High Duke could no longer be said to be the direct inheritor of these lands (1968, 90).

Second, I have previous experience with interactive fiction through a digital medium. One of the largest endeavors I have ever undertaken was an interactive fiction world created on the StoryNexus platform; while it did not live up to my expectations and issues with the platform eventually led me to cancel work on the project, it is worth noting that I highly value the medium of interactive fiction, especially as it tends to relate to gaming and learning. An important takeaway from the StoryNexus platform, however, comes from an informal study conducted by its creator, Failbetter Games, who reported that despite the marketing of their *Fallen London* as a game, the majority of users polled stated that they identified themselves as either readers and gamers or readers exclusively, with only 13% of respondents identifying as a non-reader (Kennedy).

This brings me to my third and final major motivation for this project. I am an education major, and as a result many of the things I do focus on learning and pedagogy. This project is no exception, although I work almost entirely based on theory rather than practice—I am confident that someone studying an interactive fiction work will learn from it. This stems largely from Failbetter Games' research and also that of James Gee. While the average user of interactive fiction is more likely to describe themselves as a reader rather than a gamer, I feel that Gee's argument that games are (“Learning by Design” 5) also applies to the narrative format of interactive fiction. The potential educational aspect of historical interactive fiction is not what

my project focuses on, as I am more concerned with the simple act of portraying medieval life and times with a reverent amount of accuracy and engagement, rather than being concerned with education proper. Nevertheless, it is something that I considered, and some of the features of my platform, such as the ability to inject so-called “trivia” elements into the narrative, are based on adding some additional educational value to the experience.

Part of my goal in this project has been to emulate the success of less accurate historical fiction elements; while my more slavish attention to historical canon means that there will be some differences in the scope and scale of my work, as the only issue that I am willing to bend beyond known medieval parameters on is with regards to the currency system, as that is part of a crucial game mechanic. One example of this would be Ubisoft's Assassin's Creed game series, which is set in historical context and features the player taking control of a fictional Assassin; the first installment is set during the Crusades and deals with real figures, albeit by weaving them into a fictional conspiracy over the control of world-changing artifacts. Likewise, other games like the Europa Universalis and Crusader Kings series have shown that blending history with entertainment is entirely feasible, although as grand strategy games their focus is on an adult audience and they have a more abstract approach to the everyday business of the Middle Ages.

My goal is a little more humble; barring the large-scale theatrics, players experience a handful of events that are plausible and do not stretch reality; my personal experience with writing on StoryNexus has been that the nature of the interactive fiction platform is that the ability of the player to exert a degree of agency in the world overrides some of the need for overwrought dramatic narratives that would veer away from realism.

Chapter 3. Writing Process for a Historical Fiction Narrative

For the sake of practicality, the narrative is divided into Acts. These do not reflect any conventionalized measure of content or time, rather they reflect breaking points where events and locations that were available in a previous act may no longer be available, or may change dramatically (for instance, Act 1 takes place in a single day, and Act 2 takes place the following night). This is mostly done for the sake of the XMICAE platform, which I will discuss later. Each act was planned with a similar length, however, in general play of between ten and twenty “storylets” (a term I borrow from StoryNexus), each of which reflects between 50 and 250 words of written content, so each act was planned to run about 2000 words. Some acts, however, have more breadth; Acts 4 and 5 in Krakow are significantly larger than the other acts, but the player will typically not experience all of the content. Each act will also award a player with a certain number of advances, allowing them to further customize and upgrade their character. In actuality, the practical length of these units is a little more difficult to gauge; for instance, Act 1 takes around 15 player interactions to complete efficiently, but more interactions are possible, as is the case in every act; the minimum interactions to get from the beginning to the end of the protagonist's adventures is around forty-five to fifty, but this requires prior knowledge and highly economical decision making.

The intro area is not inspired by any particular place, rather serving as a space for the player to safely explore and learn about the future game systems and the user interface of the design. It begins with the player finding themselves in a village at the Polish border, at which point they go and speak with the count (they are forced to do this) to get a passport to venture further into Poland. This marks the end of Act 1. The player has the opportunity to wander a

little, exploring the scenery, and may also explore a cave area further for a chance to find a reward; the cave uses the random character-based tests. The main goal is to find the foreman, rather than wander, so this area is not particularly large. The player may then return to the count to get the passport, at which point he is offered the opportunity to join in a manhunt or simply rest in the church and wait for morning, when they leave for Krakow. This is the official end of Act 2.

Originally, I had planned a scene to add some adventure to the travel with a confrontation between followers of Leszek and a noble who supported Władysław III Spindleshanks, but this was excised in favor of realism; the two rulers were not at open war, and while their followers may fight it would be unlikely for them to do so in plain view of a traveler. Instead, I decided to replace it with a simple conversation between a traveler and the protagonist, with the traveler providing basic advice about Poland and its current state. After this is done, the player continues to Krakow; there is not much exposition given about the journey because of the fact that travel exposition would be terribly slow for the player, and much of the first two acts consists of slower elements to begin with. There was a plan to also drop the protagonist's origin here, with the traveler observing “So, you're traveling home to/through Germany [the term is used for the sake of a modern audience, who may be unfamiliar with the concept of the Holy Roman Empire, but are likely familiar with the concept of Germany]. Is there anything you'd like to know about Poland?”, but ultimately this was decided to be an extra level of detail that detracted from the intended experience of exploring Poland. My research seems to suggest that the Fourth Crusade is rather diverse, but has not turned up any specific demographics, so it was planned to have the player come from England, France, or Germany at random.

Act 3 begins when the player goes to pass through the gates of Krakow. Originally planned to be more extravagant, the focus seemed naturally to shift to a conflict between Knights of the Sword and a merchant, whose goods are stolen by a member of the order. The player is given a fair breadth in how to respond, though the option of violence is met with a potential for failure, which will set the player back for their future endeavors.

Once the player is inside Krakow, I tried to make it clear that the size and expanse of the city were large, if not massive, for the time, with an eye toward ensuring a perception of the city as being a bustling urban center. As in medieval times, signage is used as a way for the player to navigate the city, although much of the general interaction with the city is relegated into moving between large districts; this approach speeds up play by keeping the player bouncing between locations, rather than having them check street after street for their destination.

Leaving Krakow is a little bit of work for the player. With funds depleted, they must make find one of several ways to make money before they can book passage to Torun alone or with a traveling merchant. This is a place where characters will find that having a degree of versatility in their skills can be important. Particularly clever players may figure out that the church is able to provide resources to a traveler in need, while ones who exhaust every other option may wind up pursuing a mundane job or even theft. This can be one way in which the player uses up too much time, and is forced to travel in colder, less hospitable weather.

Eventually, however, the protagonist will secure passage by boat to Torun, which marks the end of Act 3 and the beginning of Act 4, which focuses on Torun and the aftermath of the player's choices.

Torun is a based heavily on conjecture, rather than the set of data points that can be used

to reconstruct Krakow; while I will not be pursuing a perfect recreation of either settlement, in favor of a more narrative flow, there are no known sites in Torun that can be traced back to 1203. Although it is settled at the time of the player's visit, it is much smaller than it would be just a few decades later, as the influx of German settlers and their trade has not yet turned it into a major city. We could safely presume that the old town of Torun's layout resembles the plans on the original city charter, as with Krakow but as they were lost in the 13th century this is of relatively little value. The player does not experience Torun with the same level of detail as Krakow because of this, but rather experiences it as a stopping point with more action-based possibilities than the location-driven elements of Krakow. Here they encounter some crusaders who are headed off to join the Fourth Crusade, and are able to have a conversation with them about the motivations of crusaders. It is from Torun that the players' characters begin their foot journey from Poland to Germany.

The largest features of Torun come in the form of its tavern and its market. The amount of time players spent in Krakow is also important; if the player decided to idle around or took too much time wandering the city it may already be late in the year, which can serve as a detriment to their further travels as supplies will be harder to come by. Once the player has reached this point they must either continue onward without the option of acquiring more funds except by selling items that they acquired in Krakow, only one of which yields a profit. Similar to such games as the Oregon Trail, they are presented with some information about their current status, their planned travel times, and their current resources. They may also choose to travel alone (which is free but requires significant skill and resources) or in a group (which is safe, but requires the player to have earned more denarii in Krakow).

From Torun on, there are a number of smaller encounters. Many are entirely based on the player's current status, and some of them are bypassed entirely if the right conditions are met. These reflect potential points of failure for the player, or other elements of travel. While these aren't particularly solid yet, they will touch on the status of the Jews in medieval Poland, as well as the increasing Germanic influence and immigration into the western parts of Poland. Other encounters may include a run-in with an outlaw or a fellow traveler, or environmental hazards such as a storm, washed out bridge, or similar problems.

This is the first part of the narrative where the protagonist can die, in part due to the fact that it is really just a test of the player's decision making and their characters' skills. At no point is this a foregone conclusion; someone who opted to make the tactically poor decision of selling everything they own for some furs and food may still make it through, but will have a more difficult time of things, as will someone who runs out of food or entirely neglects the Survival skill. That said, this is still medieval Europe, and the wilderness is punctuated with small settlements, so the main causes of death are dramatic (a tangle with an outlaw, getting caught in a blizzard); other failures of planning are not in and of themselves entirely fatal, though failures will result in a progressively poorer epilogue.

Writing in Acts

One thing to take note of is the act structure; for the most part acts are simply a way to organize my folders for files so that I can reuse common names (like "road"), and so that I can keep each object straight. Both Act 1 and Act 2 take place in the same village, and combined they make up about the same number of separate objects as Krakow does if one counts transitional files (kept in a separate directory if the file is likely to be used in two acts), and rolls (which are

kept in a separate directory as well).

With this said, each act represents a thematic shift, though this is not necessarily explicitly stated; Act 1 is highly linear, while Act 2 adds in the first player options, though it is very short. Act 3 offers the most freedom, and Act 4 has all the consequences and rewards for players' actions.

Originally, I had planned to separate the exploration and resource gathering in Krakow into two acts, but before long this was replaced by a resource system utilizing three pools: rent, denarii, and weeks in Krakow. For the most part, Krakow does not care about how many weeks have been spent; it is something available to the player when they go to rest, but the main difference occurs in Torun and other Act 4 elements. However, rent and denarii serve as an impetus for the player to continue. They are able to pick up a couple items in Krakow from merchants, which will make their journey easier, but for the most part Krakow exists to give the player a source of denarii. On the other hand, Torun presents few opportunities for a traveler to make money, and is really a money sink.

Rent has the upside of serving as a tracker for the day of the week. In the middle ages, the week is treated differently than it is in modern times; all my sources agree that each day in the market in particular would look wildly different, which of course means that each different type of craftsman would be doing different tasks on different days. Gies points out that artisans are allowed to sell only on certain days, and although these laws are often violated the general market square would follow the rules (*Life in a Medieval City* 90). Many of the events in Krakow are subject to fortune; each player will hopefully have a very different experience based on the actions they choose to undertake, and while there is a slow but persistent means of making

money as a longshoreman, the player should be able to achieve roughly twenty to forty denarii worth of income in their first week in Krakow.

Integrated Review

One of the crucial educational points of any material is its ability to be reviewed. In this sense, a dynamic narrative is more than welcome and is quite helpful from an educational perspective. The way that I opted to achieve this is to allow the narrative to be reset upon completion, restarting from Act 1. This way, the player has the ability to go through and re-engage with content, with two caveats: the player's protagonist gets stronger on each playthrough, and gets a slightly different experience on each playthrough. The first is intentionally designed to encourage different styles of play to allow someone replaying the game to experience content they did not feel comfortable attempting due to their character's game stats, and the latter is a consequence of elements of randomization and choice. While this prevents a true review in the traditional sense, it encourages a second glimpse at the content as well as providing an altered perspective on the second attempt.

Chapter 4. Integration of Game Elements with Narrative Structures

The game elements of the project are built upon a system that I refer to as the ABACUS system. ABACUS gets its name for a generic list of proficiency ratings (Agility, Brawn, Awareness, Cunning, Understanding, and Sympathy), although these are bolstered by a selection of skills that were deemed potentially useful for the player's character. Inspired by the design philosophy of Dungeons and Dragons, but with some revisions for a lighter, minimalist design philosophy, my digital implementation of ABACUS uses two random number generators that provide integers from one to twenty, simulating two traditional twenty-sided dice. This allows for a probability curve, making each bonus that the player gets from their ABACUS rating or Skills more significant than they would be on a single 20-sided die, at least within the limitations placed by the system, which include a maximum rating of 5 in any of a character's attributes or skills, which are detailed in the next paragraph.

Characters in the ABACUS system, such as the protagonist of my story, are built using a series of advances; the exact method for doing this is mathematically derived from a formula; the cost to advance an ABACUS attribute (which see more use) is the square of the desired value (ABACUS values start at a rating of 1), while the cost to advance a skill is half the square of the desired value rounding up, except for the first level in a skill which costs 4 advances. Players get a starting pool of 62 advances to customize and upgrade their character.

There is an issue with the game system as it stands in that the player is pressured to make most of the decisions about their character at the beginning of the game, or miss out on the benefits of their use in the very early segments of the game. Players are given a number of “advances” that can be used to customize their character. If they do not use them, they will be at

a disadvantage early on, though nothing prevents advances from being spent at any time. To counteract this I attempt to carefully balance out the usage of ABACUS ratings and their linked skills, so that players who have made any decision remain satisfied. There is also the potential for attribute requirements throughout the game, which allow for an underused skill or attribute rating to be used outside of the ABACUS rules to access additional content or alternate routes.

Character advancement throughout the course of events is minimal; players are awarded advancements in place of predefined character growth, and typically receive one for completing an act as well as when particularly noteworthy events occur; it is somewhat more common for characters who are successful to receive more advances, though this is not always the case, as some advances are rewarded following a thorough exploration of the protagonist's surroundings. Although it is theoretically possible to count the number of areas that a player has visited in a number of different ways, this is reflected in the potential for a number of places which are deemed to be unlikely finds to immediately reward the player with an advance to customize their character.

ABACUS' interpretation in the final project was stripped down a fair deal; I opted to make the system as simple as possible to allow the ultimate amount of narrative freedom, something which came at the cost of having a lot of gameplay elements that I had originally planned on. In addition, I removed several game elements that would have been easily communicated and familiar to the player, like health, because I felt that in the end they were more likely to interfere with the desired narrative than enable it, although the use of game elements in fringe cases was maintained for events that either maintained a desired feel for the narrative or enriched it by adding elements of risk and reward or player choice.

Chapter 5. Technical Aspects of Interactive Fiction Deployment

One of the key focuses of my project has been dealing with the practical application of both research and writing-oriented tasks into a finalized product that is ready for use both by myself but also by likeminded individuals. This has culminated into the formation of the XMICAE project, my personal software to provide simple and versatile display of content, which is made up of a number of independent modules that come together around the XMI core, which reads human-legible files and converts them into both game logic and final narrative display. A snapshot of XMICAE in action can be seen in the Illustrations section.

Human-Content Interface Philosophy

From the beginning of the project, I approached desirable platforms with a number of goals. I wanted a user-opaque system that hid most of the game logic from players, making the final user experience into less of a game and more of a narrative, even though the game elements would be available. I also wanted something that could display large amounts of text in a minimally fatiguing manner; this meant that the majority of screen space would go to text display purposes. My final criteria was that the content parsing tools should be legible to a broad audience, and that there should be only a minimal level of technical know-how required to get the files to work, and that the files should be able to be parsed through a system that would be independent of any platform-specific idiosyncrasies.

Ultimately, this led to the creation of XMICAE; while other alternatives existed and were freely available, there were a number of downsides of each. StoryNexus, my normal platform of choice, does not support user-opaque narrative or game elements, and devotes a large portion of

screen real estate to these instead of content display. A variety of other platforms I evaluated were either not capable of parsing files without their own idiosyncrasies or had a high entry barrier; the only platform that met my requirements to a satisfying extent was that of Sryth, a fantasy game, but that was not available for private use.

XMICAE for the Layperson

XMICAE is my pet term (derived from the lengthier technical name “eXtensible Markup Interfaced Customizable Adventure Engine”) for the platform that loads all the content I use. My writing about the XMICAE platform used to deliver the content of my thesis is somewhat complex. The short description is that XMICAE runs like any website, but is custom-designed to handle the game elements of the ABACUS system that made it into my project, as well as a number of other random number generator (digital dice, essentially) based game elements, and to read and output specially formatted data. It suffers from a number of idiosyncratic design principles, however, and is not generally suitable for wide-scale use; even with the technical manual for XMICAE it would be difficult to create a finished product using the platform without the degree of intrinsic knowledge that someone who created it has, which is a consequence of XMICAE being the first major programming endeavor I attempted.

One core design functionality that I built in intentionally is user opacity; XMICAE does almost everything behind the scenes—this is largely a personal choice, but for the most part the end-user does not know what is happening with regards to any particular result. There is an unused notification system in the XMICAE code that facilitates this, but typically any game elements that come up are conveyed through narration, rather than a sidebar.

XMICAE is built for compatibility across systems, which is something that it does fairly

well even if the act of creating content for XMICAE is not immediately intuitive, and its output should be (though not necessarily perfectly identical) across all major web browsers. Its simplicity in function makes it secure against attacks, and very widely compatible.

XMICAE is also open source, and is freely available. It runs any Apache web server that supports MySQL and PHP, though it is designed and most tested on Windows using the WAMPSEVER program, with secondary testing on a Linux based Apache stack.

A Computer Scientist's Approach to XMICAE

Short for “eXtensible Markup Initialized Customizable Adventure Engine”, XMICAE is a PHP-driven web application that aims for maximum compatibility across systems and a large degree of user extensibility.

Working in PHP has some benefits and some issues. On one hand, PHP is wonderful because of the fact that it is built for web design from the ground up; all of the input and output functions found in XMICAE run through HTML, with a little bit of Javascript where it is more compatible than native PHP functions. It is also secure as a game, as most of the variables are stored on the server, making it difficult to cheat or otherwise manipulate the system. This is not meant to imply that the program is ironclad; there are some minor known vulnerabilities, such as session hijacking, that were deemed to be costly to work around for the context of the project, and unlikely to occur within the context of entertainment or educational software. PHP runs on a central server, meaning that a user will not typically need to install any software they do not have to experience play in its entirety, as any relatively modern web browser will be able to handle XMICAE's output as well as the CSS it uses for formatting and style. Unfortunately, this means that the feedback that players receive has a limited degree of flexibility unless it incorporates

elements of Javascript, which has essentially the opposite set of strengths and weaknesses as PHP does. PHP also requires a central server for hosting, which could have caused problems if I did not have one for my own use, and it does have some platform-specific idiosyncrasies that can get in the way of compatibility if certain best practices are not followed.

To give a short postmortem of the project, it is highly functional and stable, but falls victim to some design flaws born out of my limited knowledge of best practices at the beginning of the project. The code that became XMICAE began about fourteen days after I took up PHP with a random number generator, and lasted for several months of massive progress in my own coding abilities. Unfortunately, a number of mistakes occurred during the early development process. I used the mysqli extension instead of PDO, which I find more intuitive, for database interactions and I avoided object-oriented programming for much of my early work on the system. The level of commenting and division of code into included files and functions is largely piecemeal, with far too much reiteration of code that should have been moved into function and while the platform works it does so in a relatively inefficient manner (in terms of code complexity; I've never experienced issues with user load). XML was also a questionable choice for data storage, as it was not intended for use as a data serialization scheme, despite the fact that it is frequently used as such. While XML works for this purpose, it is very difficult to transcribe information by hand and there are no popular and functional graphical XML editors, something which was initially going to be an integral part of the project, but has since been largely removed.

Another critically crucial flaw in the XMICAE system is the fact that it has no built-in method to fetch variables from PHP, and while it is possible for individual mechanisms to utilize

some of the more direct functions, this means that each function has to be manually configured to do so. The lack of ability to pass by reference is a concern that is crucial to the XMICAE project, and it would be difficult to implement a system to validate and parse input again, as it would require a complete re-test to validate that there were no errors created in the process. Some of this has been fixed in a per-case basis; narrations can return the player character's name through a pair of curly brackets (“{}”), and within narrations the numerical value of a state can be retrieved with a regular expression (“state^” will return the value of “state” for a character as a number in the narration output).

XMICAE is licensed under the MIT license, and an early version that my creative project runs on can be found on SourceForge. While I would advise against using it for any commercial projects due to security and extensibility concerns, it is usable. I do not know of any security *vulnerabilities* in XMICAE, but I am aware of a number of suboptimal security fixes that have caused issues with normal functionality. For instance, the solution to SQL injection is a very strict sanitation of input that will break many schemes of data serialization, including PHP's native format. The mysqli library's built in prepared statements do not like to execute in many situations, and I was only able to get them to work once despite repeated efforts.

What is important, however, is that XMICAE works fairly well. It is compatible with a wide range of systems, which was a crucial early-development goal as I supply my own web hosting for the project on a pre-configured LAMP stack, while I develop on a WAMP stack. Barring file system concerns there should be no issues whatsoever with compatibility; a few of the functions used, like PHP's rand() function, do vary based on operating system, but not within the context of likely use or to a degree that makes them inoperable. Most functions, however,

should be compatible to a high degree of precision, like PHP's cryptography mechanisms—moving an XMICAE install from a WAMP stack to a LAMP stack and plugging it into the same database will only cause errors if they are created by PHP. Some functions that would work on both of my setups, like HTML header functions, are avoided for the sake of compatibility, instead being replaced with Javascript alternatives.

When storing user passwords XMICAE uses Blowfish encryption with a salted hash; it is the only open-source game/IF project that I am aware of to support such a powerful level of encryption by default. As a result, even if the database is accessed unlawfully user passwords will not be compromised, as a result of the one-way hashing process. It is possible to further secure these hashes by salting them with the username associated with an account, which ensures that the hashing process always functions identically. While this is not the most cryptographically secure method of salting hashes it does mean that a user who registers for accounts with commonly known passwords and then compromises the database will not be able to immediately compare their accounts' passwords with those of other users, which protects users in case of a breach. Other information is not protected, but since XMICAE stores only usernames and e-mails, this is unlikely to result in any significant damage to users in the case of a leak of information; usernames are typically not regarded as being sufficiently sensitive to warrant encryption, and hashing an email address would defeat the purpose of asking for it in the first place.

The core functionality of XMICAE comes in the form of its XML parser and subsequent display. This takes the form of three discrete components: an XMI master file and two included components, the XMI Include and Action (I capitalize the code module that handles game

attributes, and refer to the XMI component in lower-case) systems. The XMI system handles request authentication and generates final prompt data for the purpose of saving user progress (through the narrative; other forms of progress, such as the ABACUS system, are saved as they occur).

Data storage for player information is done on a number of tables in a single SQL database, most of which are globally indexed with a common user ID that is referenced when they register an account. As the inventory system was abandoned in favor of a more narrative approach, there is no storage of reference tables of any sort. States, tokens, and stokens are all arbitrarily defined and referenced only in play; as the game state is not user-transparent this is not an issue. To clarify that last statement, some IF platforms such as StoryNexus have a user-transparent system that informs users about the current narrative state by allowing them to visually review all the XMICAE item equivalents that they have received. XMICAE opts for a more traditional game/IF distinction, and only makes game elements visible. IF elements are entirely hidden in XMICAE's internal functioning, which makes debugging slightly more difficult (anyone with access to the database can, however, manually review game states for errors, allowing for offline testing or emergency fixes). Plans to incorporate administrative functions into XMICAE itself met with issues with the code, and it was decided to scrap them in favor of keeping the platform light and avoiding potential security risks that would come with the adoption of such a system.

Data storage for game content is handled through loose .xml files. This is both problematic and beneficial; on one hand it is a remarkably simple solution compared to some of the other possible ways of handling data storage. The predominantly useful element of this is that

it is possible to do with the file-system things that would require some intricate database handling. In addition, file-system handling allows for easy manual override of files on a server using a File Transfer Protocol client, something which is useful for test environments, and on a native client like my testing machines one can simply drop a file into the directory the usual way, keeping them open in any XML editor and continuing to tweak them without having to go through any additional access methods.

Any file-system accessing method of data access has some potential security vulnerabilities, however. It is theoretically possible to specify an arbitrary XML file to load from the server, including using directory overrides to access content that is not within the XMICAE configuration. This is only possible for files with a .xml extension; the platform is hard-coded to append that file extension by default and only a modified version could load arbitrary file types. Adding a further degree of complication, this would only happen if a user was able to gain unauthorized access to XMICAE's game data files; these directories are typically read-only (if they are even available at all) to the general public, and could only be modified by someone who has access to the server's file system and the XMICAE platform directory; at this point XMICAE is not the major security vulnerability. This could become a security vulnerability if XMICAE were to be upgraded to include administration functions that included in-platform content creation, but this could be solved refusing to allow additional directory selection characters in both the directory and file input fields in the XMI parser. Even further, the severity of this potential security vulnerability is inherently limited because XMI only returns XMI-valid elements of XML files, not the file itself. Unless sensitive data was stored in a user-transparent form (which requires some very specific combinations of tags that are unlikely to occur naturally

in another system), it would only tell the user whether a file exists. It is not a meaningful security vulnerability now, and it is unlikely to ever become one unless significantly more features are added to the platform without some minor tweaks being made.

XMI uses a prompt system that utilizes HTTP POST forms for a fair deal of security. Although XMICAE is not able to intelligently respond to user input in real-time, it does not need this. Instead the platform responds to user input when a request is made, and holds that information until it is fetched by a prompt. These requests are stored in PHP's session variable cache, a collection of data that is associated with a particular user and that is only flushed when certain conditions are met; this is typically limited to either a script requesting that the user's login be terminated, as in the "logout.php" code, or after an amount of time has passed which is influenced by global PHP settings on the server it is associated with and is based on inactivity and server load. Each prompt carries type and ID information; type information is actually based on the file that XMICAE is to execute (for security reasons, these are hard-coded, consisting of the Roller, XMI, and Action scripts only), and the ID is simply the number that matches the desired prompt. HTML form elements are then created that the user can click on (these are essentially button analogues, but appear as blue text similar to a hyperlink) to send the prompt type and prompt ID to be executed back to the server. It is possible to "hack" XMICAE, or create an automatic player for XMICAE, by altering the form data by the end-user, or by intercepting the prompt in the network. XMICAE stores an additional list of valid prompts in some contexts, to prevent old prompts from being called, as it does not unload prompts automatically due to potential errors that could be caused by the deletion of these prompts; PHP will automatically clear out session variables if it is running out of memory, which, combined with the relatively

small memory footprint of prompts (a matter of bytes) means that the old prompts are unlikely to cause any harm to system performance.

There are three types of prompts used in XMICAE: XMI prompts, Action prompts, and Roll prompts. XMI prompts load up XMI data. This XMI data is stored in .xml files with prompts specifically tailored to the XMI system. Action prompts contain specific information about functions to call that write data to the database; these are only executed once if the system is properly functioning, and allow for changes in game state (except for remembering files to load in case of session termination, which is handled within the XMI parser itself for maximum security). They utilize entirely separate code from the XMI parsing system, as actions are never reflected in real time to the system. Roll prompts are a special case, containing both XMI prompts and action prompts as well as roller initialization data that is unique to Roll prompts. There is also an unused prompt type, Notification prompts. As of September 2014, no XMICAE elements actually create and utilize Notification prompts; they are the user notification counterpart to Action prompts, and include image display capabilities as well as traditional text display functionality. All prompts are communicated through either an “action” or “roll” element in XMI, which will contain XMI data and may contain action data.

An upside of this system is that XMICAE uses relatively little bandwidth; while the difference between sending the full prompt information and the limited information in the form is quickly eclipsed by the amount of content being sent to the end-user, this optimization could help users on extremely limited connections (either by speed or maximum data volume) get the best experience from XMICAE.

The XMI Include system handles the final file parsing as well as some light database

interactions. It calls functions that validate database information, typically in the form of a direct comparison, to ensure that an XMI file can be shown. This is a distinction that can be made several times throughout a file; it is not good form to have an XMI file that loads without any options to be shown, as the user will become stuck staring at a blank page, but individual objects can be shown or hidden. The XMI Include functionality evaluates each XMI file as a series of objects, and then runs through each object, displaying objects in a set order. This means that it is possible to achieve any formatting goals through the clever use of a variety of objects, though all the code called within objects is based on further elements, such as titles, narration, actions, and rolls. The exact features of the XMI Include functionality is beyond this commentary, and it can be found in the XMI reference documentation, but the XMI Include code is the part of XMICAE that handles the end-user experience, though it does not write information to the game database except for serialized XMI prompt data, which is written using a prepared statement at this point to allow for security in case the Action handler receives a terminating error.

The Action system writes game data to the state, token, and stoken databases. It is not typically used for gathering information; the XMI parser does variable checking. Each of these databases has its own role; tokens and stokens are boolean (true/false) elements that are stored in an index with a player's ID. A token or stoken that is present evaluates true, while a stoken that is not present evaluates false. The difference between tokens and stokens is a matter of persistence; as tokens are not stored in a database with a primary index, it can take more time for the server to read them, so the two systems exist so that the majority of tokens that are used only in specific cases can reside on the stoken table and be wiped away to save time on future database references and reduce memory usage, while tokens that will persist between acts and locations

can be stored permanently for the duration of a player's experience. Tokens can be deleted on an individual basis, as can stokens, but all stokens that are stored in relation to a specific player ID can be wiped using a single function. States are a little more complex; they are tokens or stokens that can have a count associated with them. This functions as one would expect; a missing state is equivalent to 0, and states can be positive or negative. States are created and modified using the same function; a state that returns to 0 is maintained in the database, however. States can be deleted on an individual basis like tokens. Universal state and token destructors for an individual player do exist, but are only used to allow a player to restart the content and experience a different path.

The Roller system handles most of the game logic all at once; it is little more than a comparison system that hooks into a list of random number generators (in “roll.php”) to provide a pass/fail outcome for most events. It also allows the player to specify events that occur regardless of failure or success. The Roller's output consists of the same sort of XMI output that any action can create, but does so in triplicate for the success, failure, and general outcomes. It then sends only the success and general outcome prompts or the failure and general outcome prompts.

Note that while I have detailed the ABACUS system, the roller can support a number of other game systems; there are default values provided in “config.php”, but while there is a specialized ABACUS random number generator among the “roll.php” functions, there are also a number of random number generators that can be used for alternate rulesets or even for an arbitrary number of arbitrarily defined dice, including such potentially wacky combinations as three dice with four hundred and ninety six sides each. While XMICAE uses the less random

PHP `rand()` function instead of a more mathematically accurate randomness, its results are sufficiently random for the context of play; as the `rand()` function reseeds between each call there is little danger of inappropriate duplicate results.

Chapter 6. Reflections and Future Applications

Reflecting upon this project, there are a number of things that stand out. First, for any commercial or large scale educational application, there would need to be a major user interface redesign. Nothing I did throughout the course of the project was really centered around making the experience pretty for the user, and the lack of any visual attention grabber likely would make it almost impossible to market my work not only to educators but also to students.

From a historical research perspective, there were some mistakes made in the selection of topic. While it is possible that there are many good Polish texts on the subject of the early 1200's, I found only a limited selection of English-language texts, and as such I was searching for tidbits of information with a lot of repetition as I utilized works by a limited pool of authors. For such a large scale project, this meant that the tie-ins between the narrative and Poland were very limited, potentially limiting one of the project's goals of building student connection to the time period. Despite the difficulty of finding research that gave me enough to create a really vibrant view of life in medieval Poland, I feel that this was still a largely successful part of the project.

Educationally, it is hard to call the project a success or a failure. I was able to increase information density, but the lackluster user interface fails to engage the audience. The interface, however, would be simple enough to improve, and the groundwork for solid educational features has already been laid.

The game elements of the work were somewhat under-used; despite having a role at several stages during the experience. The desire for near total transparency made them a sort of extra feature; omitting them would have made content and code creation a lot more efficient, but

the user rarely sees them, even when they are working.

The first code that went into the XMICAE platform was written a matter of days after I had first learned to write code at all, and that many of the decisions that I made about what I was going to use reflected a novice's need to use well-documented features that I had access to working examples of. As I became more proficient with the process of writing software new opportunities opened up, and the project suffers from my late adoption of object-oriented approaches that would have allowed me to not only work in more features but also apply libraries that have fewer idiosyncracies than some of the ones XMICAE uses.

From a design perspective, I find it intriguing that I was able to implement my original vision for the mechanical function of the platform as well as I did; there were definitely features that got removed both for stylistic and practical reasons, but many of these were removed because I realized that they were based on emulating something that I had seen once instead of doing what had really been my intent all along. The only missing feature that I really wish I had been able to make is a working in-platform XML editor.

XMICAE shows that it is feasible to create an interactive fiction platform for the retelling of historical events, though the effort does need to be approached from a perspective that is conscious of all the potential pitfalls of creating a large application and finding research for the topic.

Illustrations



Fig. 1.

Example of XMICAE User Interface

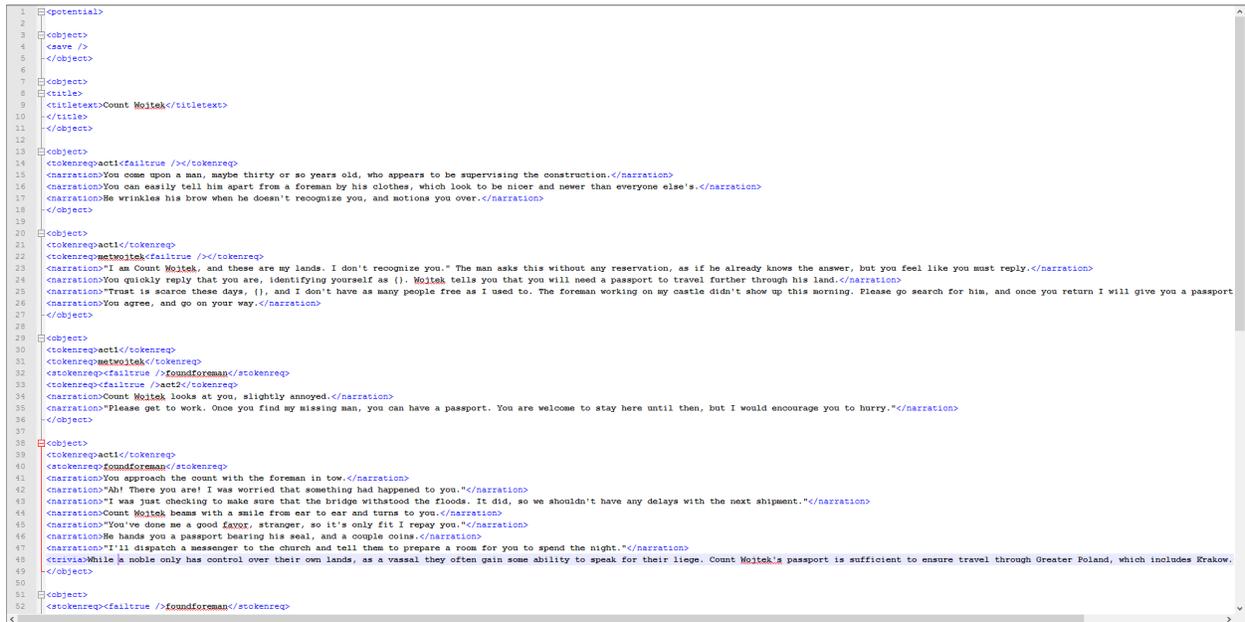
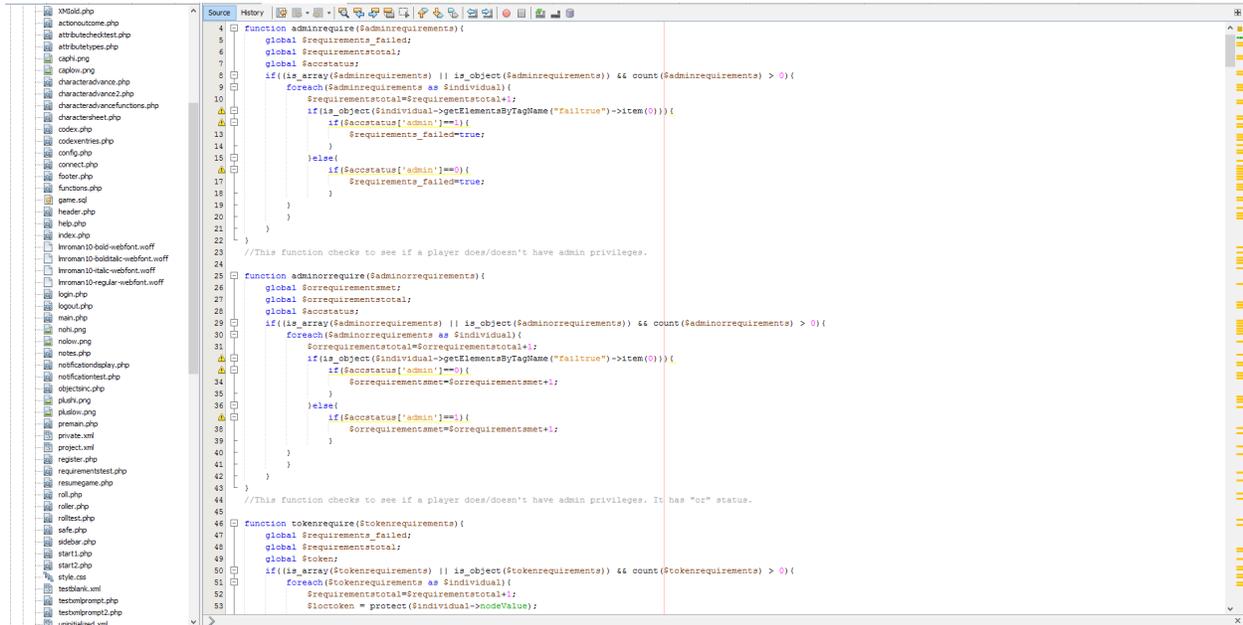


Fig. 2.

Example of XMI-compatible .xml file contents.



```
4 function adminrequire($adminrequirements){
5     global $requirements_failed;
6     global $requirements_total;
7     global $accstatus;
8     if(!is_array($adminrequirements) || !is_object($adminrequirements) && count($adminrequirements) > 0){
9         foreach($adminrequirements as $individual){
10             $requirements_total=$requirements_total+1;
11             if(!is_object($individual->getElementByTagName("failtrue")->item(0))){
12                 if($accstatus['admin']==''){
13                     $requirements_failed=true;
14                 }
15             }
16             }else{
17                 if($accstatus['admin']==''){
18                     $requirements_failed=true;
19                 }
20             }
21         }
22     }
23     //This function checks to see if a player does/doesn't have admin privileges.
24
25 function adminorrequire($adminorrequirements){
26     global $orrequirementsmet;
27     global $orrequirements_total;
28     global $accstatus;
29     if(!is_array($adminorrequirements) || !is_object($adminorrequirements) && count($adminorrequirements) > 0){
30         foreach($adminorrequirements as $individual){
31             $orrequirements_total=$orrequirements_total+1;
32             if(!is_object($individual->getElementByTagName("failtrue")->item(0))){
33                 if($accstatus['admin']==''){
34                     $orrequirementsmet=$orrequirementsmet+1;
35                 }
36             }else{
37                 if($accstatus['admin']==''){
38                     $orrequirementsmet=$orrequirementsmet+1;
39                 }
40             }
41         }
42     }
43     //This function checks to see if a player does/doesn't have admin privileges. It has "or" status.
44
45 function tokenrequire($tokenrequirements){
46     global $requirements_failed;
47     global $requirements_total;
48     global $tokens;
49     if(!is_array($tokenrequirements) || !is_object($tokenrequirements) && count($tokenrequirements) > 0){
50         foreach($tokenrequirements as $individual){
51             $requirements_total=$requirements_total+1;
52             $lootoken = protect($individual->nodeValue);
53         }
54     }
```

Fig. 3.

Snapshot of XMICA source code.

Historical Glossary

Fourth Crusade: Begun in 1202, the Fourth Crusade is only followed through 1203 in my narrative, at which point it has resulted in the sack of Zara (or Zadar; the former appears to be the contemporary noun), a Christian town under Hungarian rule.

Knights of the Sword: A sect of hedge knights associated with the crusades, similar to but distinct from more famous organizations as the Knights Templar that had been founded shortly prior to the 1203 date that my project examines (Halecki *Borderlands of Western Civilization: a History of East Central Europe* 81-82).

Krakow: Krakow is the central seat of power in Piast Poland. It contains many notable landmarks, such as St. Adalbert's Church, and Wawel, a massive fortification and complex that served as the hub of Poland's national power.

Leszek I Biały: Leszek the White is the king of Poland at the time of the narrative; while High Duke, a figure more or less analogous to the traditional definition of a king, the province directly surrounding Krakow was actually out of his control (Gieysztor *History of Poland* 75; 2nd ed.). He is a member of the Piast dynasty.

Torun: A minor outpost at the time of the narrative, Torun blossoms under the influence of incoming crusaders and Germanic settlers, who build the settlement into a trading hub for Poland.

Władysław III Spindleshanks: Władysław III Spindleshanks is one of the more influential noblemen at Leszek's time; he controls the provinces that contain both Krakow and Torun.

Technical Glossary

***AMP:** Combined with a number of prefixes, this is the technical term for a family of web-server setups that incorporate a base operating system such as Linux, Windows, or Macintosh (LAMP, WAMP, or MAMP, collectively); it is also used as a trade name for a number of out of the box distributions, such as “WAMPSEVER”. Typically the -AMP stands for Apache, MySQL, and PHP, which is the case for setups that can run XMICAE.

Client: A client is an end-user of a server-hosted web application; for the context of XMICAE most of the clients are user's web browsers.

CSS: Cascading Style Sheets contain additional formatting information that typically consists of the graphical methods of display

HTML: Hyper Text Markup Language consists of a number of tags inserted into a document to format it for display on digital devices; this standard has evolved over time and is the predominant method of displaying content to a web browser.

HTTP: Hypertext Transfer Protocol a networking method that is used to communicate with HTML files and serves as the bridge between the clients and the server.

PHP: A programming language intended for use as a server-side scripting language that allows for a number of networking functions. PHP is a popular software for running online applications, such as Google's search engine.

XMICAE: A shortened name for the platform that I created in pursuit of an appropriate tool to build my project on, from “eXtensible Markup Interfaced Customizable Adventure Engine”. It utilizes PHP to perform game and interactive fiction functionality for a wide audience.

Works Cited

- Davies, Norman. *God's Playground: a History of Poland*. New ed. Oxford: Oxford UP, 2005. Print.
- Gee, James Paul. "Learning By Design: Good Video Games As Learning Machines." *E-Learning* 2.1 (2005): 5-16. Print.
- Gies, Frances, and Joseph Gies. *Life in a Medieval Village*. New York: HarperCollins, 1991. Print.
- . *Life in a Medieval Castle*. New York: Harper & Row, 1979. Print.
- . *Life in a Medieval City*. New York: Harper & Row, 1981. Print.
- Gieysztor, Aleksander. *History of Poland*. Warszawa: PWN, Polish Scientific, 1968. Print.
- , Aleksander. *History of Poland*. 2d ed. Warszawa: PWN, Polish Scientific, 1979. Print.
- Halecki, Oskar. *A History of Poland*. 9th ed. New York: D. McKay, 1976. Print.
- . *Borderlands of Western Civilization: a History of East Central Europe*. 2nd ed. New York: Simon Publications, 1980. Print.
- "History of Krakow." *Magiczny Kraków*. City of Krakow. Web. 13 Nov. 2014. <<http://www.krakow.pl/english/5423,artykul,history.html>>.
- Kennedy, Alexis. "Failbetter Games Community." *Survey on Fallen London*. Failbetter Games. Web. 13 Nov. 2014. <<http://community.failbettergames.com/topic1151-survey-on-fallen-london.aspx>>.
- "Krakow Old Town." *Google Maps*. Google. Web. 13 Nov. 2014.
- "Medieval Town of Toruń." *UNESCO World Heritage Centre*. UNESCO. Web. 13 Nov. 2014.
- Munro, Dana Carleton. *The Fourth Crusade*. Philadelphia, Pa.: Dept. of History of the U of

Pennsylvania, 1896. Print.

"The Historical Setting: Early History Until 1385." *Info-Poland*. University of Buffalo. Web. 13

Nov. 2014.