**Overview (with relevant design decisions)**

Once we thought up the initial idea for our game, we jumped straight into the Lens of Infinite Inspiration. Although a lot of the inspiration for our game came from other games, such as Mass Effect (dialogue/meaningful choice) and Journey (moving story), we also decided quickly to pull inspiration from our own experiences. We thought that if we had common experiences as kids, players would also have the same experiences (or at least similar ones) and be able to connect/empathize with the story more. Empathy enables the player to project themselves onto the boy and give the advice they would want to receive, so we tried to work that in. An idea for allowing the player to choose gender/features to enable projection even further was introduced, but that involved a LOT more work and was discarded pretty quick.

While designing the aesthetics, we considered the Elemental Triad. Our story was most important, so we wanted to graphics to reinforce that. I think we managed to get a creepy vibe with the boy’s appearance, even if we didn’t get to include the subtle changes that reflected his state of mind. We believe the mechanics also helped, since as a bear (and an imaginary friend, at that) the player cannot physically change anything in the world, and can only influence the boy.

Once thing we really wanted to include was some continuity in the images, to kind of hint that the player’s choices were remembered and had an impact. For example, after the player tells the boy to “try hard”, the boy gets a black eye as shown:



In the following scene, they black eye and cut can still be seen, although it has faded.

****

One thing we really wanted to do from the start was display the dialogue in a “typewriter” fashion, where every sentence is printed letter by letter. The idea was also to have sound play to imitate the boy talking, but we weren’t sure how to add that into our typewriter method (see appendix) without playing it on every letter, which sounded too chaotic and weird. We figured there had to be some way to pause the game for a bit while the player was well, playing, and finding Thread.sleep() really helped. We went on to use it for more than we thought, including delays between scenes or the boy’s sentences. Overall, our typewriter method worked well. It seemed to help the player focus on the story a little more instead of just skimming what the boy was saying, and since the story was on the more important side, we thought this was a good thing. It also set the text to scroll down as it was appended so it didn’t distract the player and/or cause any confusion about why the text wasn’t showing, while it was further down in the scroll pane.

The attachment factor was added in later on, but especially helped when determining endings. We had always had the idea for it (it kind of stemmed from inFamous, which had the morality meter) but we weren’t sure how we wanted to put it in the game. In the end, we decided not to show explicitly how much attachment was accumulated because the player could kind of guess from what they had been responding with. Depending on what dialogue choices the player responded with, attachment would go up or down and it helped to take the game to multiple endings.

**Comparison to original proposal**

Our final product was extremely close to what we originally proposed, all things considered. The amount of detail we initially wanted to include was reduced significantly and the story was shorter than planned, but our core game is what we envisioned in December. We decided to keep the story a bit shorter because with multiple endings, a shorter game is easier to replay with different choices. The exclusion of the more subtle details in the end were a result of running out time, regrettably.

Probably the biggest change between our proposal and our final game is that we switched to Java after the midterm. When we were working with Game Maker, every time we solved one problem there would be another one that would take even longer to figure out. I think that for the kind of game we were making (primarily based on a dialogue system), most of the tutorials and help were directed at more advanced users and incorporated into RPGs. When we did eventually switch to java, our game started progressing *much* faster.

**Successful design decisions**

**Simplicity**

Games can be great ways to explore fictitious worlds and tell great stories. Our goal was the latter, to tell a story. Committing to telling the best story we could with this game allowed our design to be very simple. We want a story that the player can influence, so that’s exactly what the player can do. Indeed, that’s the only thing they can do. The player can only control what the Bear says to the Boy. It’s only one action, but it shapes the whole story. Of course, giving the player the ability to write their own responses would be a design nightmare, so there are only scripted responses. However, these responses do have impact and indeed influence the story. Since the game is based around the player having a dialogue with the Boy, there is also constant feedback (much like a real conversation).

**Infinite Inspiration**

Choosing to have a grounded story helped us a lot in the writing process. We do have a fantastical element to our game in the form of a talking bear, but the situations are human. Every kid has an imaginary friend at some point in time. We were able to take inspiration from our own past experiences to decide what we wanted to write. Even better, our situations were relatable enough that we could easily come up with solutions to the story’s presented conflicts. This made the writing process fun and enjoyable, and it would be easy to continue the game’s story if we so desired.

**Meaningful Choice**

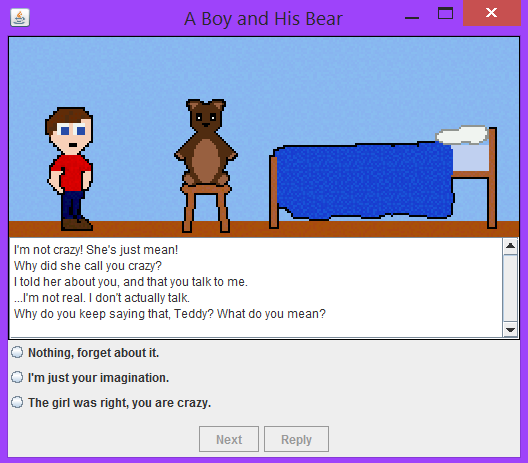
Our game originally proposed to revolve around meaningful choice. This had to be implemented before any code was written. So, our games’ script outlined how choices would deviate. The game has three achievable endings, and multiple paths that lead to them. The obvious player impact on our story is seen in Scene 3. Depending on what the player said to the Boy at one specific moment (See Appendix: Story Tree), Scene 3 can happen one of two ways. Besides that, the endings are the only other major deviations. The player may choose one dialogue option that temporarily deviates the story but more often than not, the story converges. In order to make these choices meaningful, they had to resonate with the player since we could not allow infinite deviations. This does come naturally from the Boy himself. Empathy, or lack there of, makes the story interesting and is what motivates the players’ choice.

**Judgment**

A lot of games with score or rate the player based on there performance in the game. This is one of the last additions we made to our game. It couldn’t be score based, but we did want to judge our player in one form or another. At first, we wanted to be brutally honest with the player, essentially telling them that they are a terrible person. This could have worked if we had time to make a larger game where our players could impact the Boy’s life over a longer period of time. Since that was not an option, a more subtle approach was adopted. We created epilogue screens to give a little detail into what happened at the end of our game, the followed this with a quick “Player Analysis” screen. This screen gives the player some perspective on how they arrived to this ending. So, if the player was being over protective and too attached to the boy (due perhaps to the need to always be the hero/good guy in games), they were notified of their potentially dangerous behavior. These analyses aren’t meant psychoanalyze the player, but that could be an option if the game were longer.

**“Not Real” Scenario**

Early on, we knew we wanted to give the player the ability to be completely honest to the Boy. To achieve this, the player can let the Boy know that he is just an imaginary friend. In the Scene 2, there is one dialogue option where the player can tell the Boy that he is not real. The Boy then laughs and ignores it (See Appendix: Story Tree). The game will remember if the player chose this and present it again in Scene 4. This time, the Boy is distraught and the player can continue to reveal the Bear’s true nature to the Boy.



This honesty adds an extra layer of complexity to our game’s story. We wanted to add some more flags for continuity (apart from the few we had), but couldn’t find places that would make sense and still fit in the short story we had.

**Cheap Animation**

Our game has animation in it, and it was easy to do. The software used to create the game’s art can also create GIFs. Once we realized that Java supports this image type, the idea formed. After a bit of experimentation, a simple scene of the boy blinking was created and implemented with complete success. The simple action of blinking is enough to bring the boy to life. So, *idles* were created for every scene just so we had this touch of life to our game. It didn’t take that much time, and it has a much better impact than a static image. GIF animation also allowed us to fade-outat the end of certain scenes. GIFs were not the only technique used; Scene 2 opens with the boy waking up. Switching between two images (the boy sleeping and awake) with a small time delay created this effect.

**Failed design decisions**

**The Title**

This was probably the hardest thing to agree on. The Java code was codenamed “Project Boytoy” as a joke. That’s probably the only title we both agreed on, and it was a project codename. The title “A Boy and His Bear” was our fallback title if we couldn’t find a better one in time for the demo sessions. It’s not a terrible title, but going forward with this game will require a better one.

**Working in Game Maker**

This was probably the single biggest mistake we made during this progress. Though we’ve already mentioned this, it’s worth repeating simple because so much was learned from this. Working with Java allowed us to use more techniques we already knew. Also, support for our problems now became easier to access with textbooks and the vast knowledge of the Internet.

**Sound**

This wasn’t technically a failure, just something left out. Music was recorded for the game. It could use some touch up and quite possibly a complete rerecording. However, it was left out of the demo version of the game due to time. For what we wanted the player to experience, it was not a significant addition. Also, the Game Maker build had sounds attributed to the typewriter output. This is a classic touch that would have been great to add but, much like the music, was left out due to time.

**Player Replies**

We wanted to add the player replies into the dialogue box so that they could scroll back up and go over what had happened in the conversation already if they needed to. We did so, but we quickly realized that it might create some confusion between what the boy said and what the player said. We didn’t want to preface each statement with ‘Boy:’ or ‘Teddy:’, so we left it as is and assumed the player would have enough short-term memory to remember what they had said. After, we thought that maybe we should have made the player responses bolded or italicized to differentiate them, but of course it was too late.

**How our design would continue**

If we were to continue developing our game, we would probably switch again to another game engine (be it Game Maker, Unity, whatever) but take the time to fully understand the basics of the engine and gain some experience working with that engine. We would finish the story, keep the diverging then converging format we have right now, but possibly end the game at different times based on replies the player chooses. We’d also include more detail in the graphics (such as posters, possessions, and changing colours) and improve the animation; for example, the boy currently jumps around the room and we’d enable him to walk across the room.

An idea that we’d like to implement is to do some research into personality typing. We would edit the dialogue to apply more towards different personality types, and add a pseudo-psychological angle to the player analysis at the end. Of course, it’s a bit hard to pass judgement after a short game with fewer choices, so that ties back in to extending the length.

**What would we do differently?**

I think in our next project, we’d be less stubborn. If we had accepted that Game Maker wasn’t the optimal program for us a bit sooner, we might have been able to add in the details we wanted, or extend the story a bit more. Otherwise, we’re pretty pleased with what we have going so far. We would continue it in the same manner we’ve been working on it so far, and hopefully we would have a bit more time to put into it.

**Working with Git**

Our experience with Git was interesting; there were a few mishaps where we lost some work and needed to debug a branch or two, but overall it suited our needs.

**Endnote**

We invited our friend over to try out our finished game. This was the first time we showed the finished product to someone, so we were fairly nervous. That went away shortly after he started playing. “Oh… I can’t go back… Shit” he said. This was the moment he realized that these choices do have impact, and it was quite interesting seeing the amount of hesitation he put into his choices from that point on. For us, that was validation to our game’s design. During the demo sessions, it was quite interesting watching others play it. There were some who looked so focused and then others who would play with their friends and choose as a group effort. Some took the story seriously and played honestly, others joked and picked the worst answer. Either way, they all played to see what would happen. The story made each player curious. Our simple design was indeed enough to engage players.

**Appendix: Source Code**

Example statement: *typewriter*("I'm going to name you...Teddy.");

Method:

**public** **static** **void** typewriter(String dialogue){

//reset to defaults

*defaultValue*();

*display*.append(" ");

//displays string letter by letter (typewriter effect)

**for** (**int** i=0; i<dialogue.length(); i++){

*display*.append(dialogue.substring(i, i+1));

*display*.setCaretPosition(*display*.getDocument().getLength());

//try and catch for Thread.sleep, in case of errors.

**try** {

Thread.*sleep*(40);

} **catch** (InterruptedException e) {

System.*err*.println("Something went wrong");

}

}

//enables next button again to proceed with dialogue

*next*.setEnabled(**true**);

//sleeps, checks if needs to be paused and sleeps again if true

**while** (*pause* == **true**){

**try** {

Thread.*sleep*(50);

} **catch** (InterruptedException e) {

System.*err*.println("Something went wrong.");

}

}

//newline for splitting up sentences

*display*.append("\n");

}

**Appendix: Story Tree**

