

Institutionen för kommunikation och information
Examensarbete i Medier: Dataspelsutveckling 30hp
C-nivå
Vårterminen 2008

Ittle Dew

Kompakt leveledesign som heter duga

Joel Nyström

Joel Nyström 198203304632
Handledare: Anette Lundin
Examinator: Ulf Wilhelmsson

Ittle Dew – Kompakt leveledesign som heter duga

Examensrapport inlämnad av Joel Nyström till Högskolan i Skövde, för Kandidatexamen (B.A.) vid Institutionen för kommunikation och information. Arbetet har handletts av Anette Lundin.

03 06 2008

Härmed intygas att allt material i denna rapport, vilket inte är mitt eget, har blivit tydligt identifierat och att inget material är inkluderat som tidigare använts för erhållande av annan examen.

Signerat: _____

Ittle Dew – Kompakt leveldesign som heter duga

Joel Nyström

Sammanfattning

Denna rapport beskriver två nya begrepp inom speldesign; kompakt leveldesign och planerad sequence breaking. Fokus ligger på kompakt leveldesign, och har definierats som att innehålla följande tre element: Hög rumslig densitet, lager, och backtracking. Planerad sequence breaking definieras som en mer subtil variant av alternativa rutter. Begreppen står som dels underlag för fortsatt diskussion om ämnet, då de inte förekommer i tidigare litteratur, och dels som ett sätt att förhålla sig till leveldesign som designer.

Vidare beskrivs spelet Ittle Dew, vilket har skapats tillsammans med Daniel Remar i avsikt att implementera ovanstående koncept. Speltester visar på hög grad av nöjdhet hos spelare, och implementationen anses således ha lyckats väl.

Författaren uppmuntrar till användandet av kompakt leveldesign i andra verk, med förbehållet att inte tumma på mer traditionella riktlinjer.

Innehållsförteckning

1	Bakgrund.....	1
1.1	Syfte	1
1.2	Avgränsning.....	1
1.3	Samarbetet.....	1
1.4	Ramverket, spelet Ittle Dew.....	1
1.5	Begreppsbildning	2
1.6	Kulturellt och nutida sammanhang	2
2	Begreppsdefinitioner.....	3
2.1	Vedertagna begrepp	3
2.1.1	Sequence breaking	3
2.1.2	Pacing.....	3
2.2	Nya begrepp.....	4
2.2.1	Kompakt leveledesign	4
2.2.2	Planerad sequence breaking.....	4
3	Arbetsprocess & Metod	6
3.1	Arbetsplanering.....	6
3.2	Arbetsmetod.....	7
3.3	Utdrag från projektdagboken	8
3.3.1	Förproduktion	8
3.3.2	Tidig speltestning.....	9
3.3.3	Nytt designbeslut: Varje våning har unikt utseende	9
4	Presentation av verket.....	11
4.1	Ramverket Ittle Dew	11
4.1.1	Historien.....	11
4.1.2	Spelmekaniken.....	11
4.2	Exempel på kompakt leveledesign	11
4.3	Exempel på planerad sequence breaking	12
4.4	Exempel på konventionell leveledesign	14
5	Diskussion.....	16
5.1	Motsättningar mot konventionell leveledesign.....	16
5.2	Leveledesign är inget avskärmat arbetsområde.....	17
5.3	Vikten av tidig konkret leveledesign	17
5.4	Samarbetet – intressekonflikter och kompromisser	18
5.5	Vikten av tidig speltestning	18
5.6	Problem med pacing och spelomfång	19
6	Slutsats	20
	Referenser	21

Bilaga 1 - Spelet Ittle Dew (Körbar fil)

Bilaga 2 - Pacingschema (Photoshopfil)

Bilaga 3 - Rumslayout (Photoshopfil)

Bilaga 4 - Designdokument för spelet Ittle Dew (Wordfiler)

1 Bakgrund

I denna rapport reflekterar jag över mitt arbete med att skapa en kompakt leveledesign till det spel jag och min partner Daniel Remar tillsammans har skrivit designdokumentet till och han programmerat. Jag går i detalj in på vad kompakt leveledesign innebär, och stämmer av mitt arbete mot denna definition. Jag reflekterar också över själva samarbetet med Daniel.

1.1 Syfte

Mitt syfte är att ge en definition av kompakt leveledesign och ett implementerat exempel av detsamma i ett komplett spel. Jag undersöker om kompakt leveledesign finns beskrivet i tidigare litteratur, under samma eller något annat namn. Om inte, står min definition som en grund till vidare forskning och diskussion.

1.2 Avgränsning

Jag har skapat leveledesignen till mitt och Daniel Remars ramverk, spelet *Ittle Dew*. Jag har inte begränsat min leveledesign till någon särskild skola eller ens till mina egna begrepp, men har om möjligt hållit fokus på kompakthet. Där återkoppling från testanvändare har visat att kompaktheten eller någon annan aspekt inte har fungerat eller skapar förvirring, har jag prioriterat tydlighet och spelbarhet framför kompakthet.

1.3 Samarbetet

Jag och min samarbetspartner Daniel Remar bestämde oss för att göra ett spel ihop under arbetet med vårt examensarbete. Våra respektive inriktningar var inte helt klara från början, t ex var det Daniel som till en början hade tänkt att skriva om leveledesign. Vi visste dock att vi ville göra ett spel ihop, och satte oss att spåna ihop spelidéer. Efter ca två veckor hade vi bestämt oss för ett koncept och skrivit ihop designdokument samt teknisk bibel till det. Därefter började vi arbeta parallellt med våra verk, där Daniel skapade spelet i programmet *Game Maker*, och jag arbetade med leveledesignen till detsamma. Övriga arbeten som t ex grafik har vi delat mellan oss.

1.4 Ramverket, spelet *Ittle Dew*

Ramverket är ett spel med tydliga influenser från spelen *The Legend of Zelda* (Nintendo, 1987), och *The Legend of Zelda: A Link to The Past* (Nintendo, 1992). Likheterna består bland annat i fågelperspektivet, samlandet av nya föremål som möjliggör tillträde till nya platser, blandningen av pussel och action, och en hel del rena grafiska likheter. På många sätt är det en homage till *Zelda*-serien.

Det tar även mycket influenser från spelserien *Metroid*, och då främst spelet *Super Metroid* (Nintendo, 1994). Här är dock influenserna mer subtila, och handlar snarare om att överföra vissa övergripande idéer till vår design, som t ex det jag definierar som kompakt leveledesign, som jag också försöker definiera ett begrepp kring.

Vårt spel skiljer sig också på en del punkter från våra inspirationskällor. Föremålen man samlar på sig är bara fyra till antalet, och alla fyra går att på olika sätt kombinera ihop med de andra tre. Dessa kombinationer möjliggör väldigt avancerade pussel, utan att överbelamra spelaren med för många val. Spelets upplägg och struktur skiljer sig också i det att det bara finns en enda stor ”borg”, eller värld som spelet utspelar sig i. Denna borg är dock väldigt kompakt och innehåller väldigt mycket att upptäcka i jämförelse med en normal borg i ett Zelda-spel. Se Bilaga 1 för hela spelet i körbar form.

1.5 Begreppsbildning

Mitt ledord under arbetet med både designdokumentet och leveldesignen har varit ”kompakthet”. Begränsningen till enbart fyra föremål i spelet, och att de går att kombinera på många olika sätt är ett led i detta tankesätt. Framför allt har det väglett mig i leveldesignen, och jag har även försökt att definiera ett nytt begrepp, ”kompakt leveldesign”, eftersom jag inte hittade någon tidigare litteratur i ämnet. Denna begreppsdefinition har jag också använt mig utav under arbetet med leveldesignen som en vägledning och ett sätt att stämma av att mitt arbete är på rätt väg.

Ledordet kompakthet har även lett fram till begreppet sequence breaking, som är ett väl använt uttryck inom vissa spelforum, och som jag anser kräver en kompakt leveldesign för att fungera. Sequence breaking är något som normalt sett inte är planerat av spelutvecklaren, utan snarare kan ses som utnyttjandet av buggar. I min form är sequence breaking något jag inkorporerar i min leveldesign redan från början.

1.6 Kulturellt och nutida sammanhang

För ramverket finns det tydliga kopplingar till samtida populärkultur, framför allt till spelserien Zelda, och då främst de delar i serien som utspelar sig i 2d. Även om de spel i serien som utspelar sig i 3d delar många strukturella likheter med resten av serien, har 2d-spelen en betydligt lättsammare ton, vilket också Ittle Dew har.

Huvudpersonen och likaså spelaravateren Ittle Dew är en kvinna som under spelets gång letar efter sin försvunna manliga vän. De monster hon stöter på är ungefär till hälften av manligt och till hälften av kvinnligt kön. Detta var ett helt godtyckliga val från min och min samarbetspartners sida och skall inte ses som ett sätt att jämställa spelvärlden eller ha någon underliggande betydelse.

För själva kärnverket, spelets leveldesign, är inspirationen främst tagen från spelserien Super Metroid, men översatt till en annan genre och ett annat perspektiv. Utöver denna övergripande influens är leveldesignen svår att sätta i kulturellt sammanhang.

2 Begreppsdefinitioner

Här följer definitioner av de vedertagna begrepp jag har valt att använda mig av i min rapport, samt definitioner av de nya begrepp jag introducerar.

2.1 Vedertagna begrepp

Jag har valt att behålla vissa engelska vedertagna begrepp genom rapporten utan att översätta dem. Istället förklarar jag dem ingående i detta avsnitt.

2.1.1 Sequence breaking

Begreppet sequence breaking myntades i samband med att ett onlineforum bildades kring att försöka klara av spelet Metroid Prime (Nintendo, 2002) med så låg procent samlade föremål som möjligt. Många av föremålen man samlar i Metroid Prime är nödvändiga för att kunna komma vidare i spelet, och ju färre föremål man samlar desto svårare är det att fortsätta. Spelets design förutsätter också en viss ordning på föremålen och vissa föremål är tänkta att vara obligatoriska. Detta är den planerade ruten, sekvensen. Att bryta mot denna planerade rutt kallas sequence breaking, och är alltså att betrakta som utnyttjande av buggar i spelet.

Även fast den tidigaste dokumenterade förekomsten av uttrycket är från 2003¹, har fenomenet funnits sedan åtminstone 1986 då spelet Metroid (Nintendo, 1986) släpptes. Även i detta spel finns många buggar att utnyttja för att ta sig till platser i spelet som inte är tänkta att nå vid det tillfället (med tillfälle menas här hur uppgraderad avataren är vid det laget). I uppföljaren Super Metroid (Nintendo, 1993) finns det även här många sådana buggar att utnyttja för sequence breaking, men här börjar vi också skönja en leveledesign som tillåter och är *planerad* för alternativa rutter. Man kan alltså anta att spelutvecklarna tog till sig en del av de positiva bieffekter buggarna i Metroid innebar, och applicerade dem i Super Metroid, så att avancerade spelare kan utforska vanligtvis låsta delar av spelet mycket tidigt, vilket hos många spelare ger en stor känsla av tillfredsställelse. Detta tog Nintendo ytterligare ett steg längre med spelet Metroid: Zero Mission (Nintendo, 2004), som är en nyversion av originalspelet Metroid, med ny grafik och förbättrade kontroller. Detta spel är fullkomligen belamrat med planerad sequence breaking, och det är uppenbart att detta var ett tydligt mål med leveledesignen från början.

Som en sidnotering är det värt att nämna att tiltning, ett begrepp inom flipperkulturen, mycket väl skulle kunna ses som det första försöket till sequence breaking. Det går ut att luta hela flippermaskinen så att kulan rullar till rätt mål. Sedemera lade tillverkare av flippermaskiner in en kontroll för detta som bestraffade spelaren om spelet upptäckte tiltning.

2.1.2 Pacing

¹ Se forumtråden "Ice+Grav before thardus" på hemsidan *metroid 2002* (http://www.metroid2002.com/sequence_breaking_topics_tbj_V4.0.php)

Inom denna rapport kommer jag att använda mig av det engelska uttrycket *pacing* som den medvetna ansträngningen att via leveledesignen låta spelaren uppleva spelet i en perfekt balanserad takt, där man tar i beaktning att inte överbelamra spelaren med för mycket information, samtidigt som motståndet hela tiden ligger på en utmanande nivå.

2.2 Nya begrepp

Jag har valt att introducera två nya begrepp, kompakt leveledesign och planerad sequence breaking som ett underlag till att tala om och analysera mitt verk. Dessa begrepp har jag sökt efter i existerande litteratur så som *Level Design for Games* (Co, 2006) och *Game Level Design* (Byrne, 2004), både under dessa namn och under andra namn eller del av en större definition, men har inte funnit de beskrivna i någon fullgod litteratur. Således står dessa begrepp som teorigenererande.

2.2.1 Kompakt leveledesign

Jag har definierat en lista på egenskaper som beskriver det jag kallar ”kompakt leveledesign”. Denna lista fungerade som ett verktyg för mig i mitt arbete, och hjälpte mig att kontrollera att min design uppfyller kraven. Den fungerar även som utgångspunkt för en vidare diskussion om ämnet, och för att undersöka om begreppet redan finns definierat i en annan kontext eller under ett annat begreppsnamn. Jag kom fram till följande tre egenskaper.

Rummet och dess densitet: All leveledesign syftar till att fylla ett rum med innehåll (min definition). Leveledesign är alltså starkt kopplat till utrymme, ytor, och densitet av levelement. I en kompakt leveledesign är densiteten av levelement i rummet högre än normalt. Ett levelement kan innebära olika typer av hinder, t ex ett monster som måste dödas, eller ett pussel som måste lösas.

Lager: I en kompakt leveledesign finns det i rummet flera lager av pussel eller andra utmaningar ovanpå varann. Olika förutsättningar innebär tillgång till olika lager, men det finns också direkt tillgängliga lager som är svåråtkomliga beroende på en icke uppenbar implementation. Spelaren kan alltså se på ett rum med olika ”ögon”, och upptäcka nya utmaningar och möjligheter ju mer hon utforskar rummet.

De olika lagerna är inte nödvändigtvis separerade från varann, utan existerar jämsides. Lagerna är alltså endast lager i abstrakta termer, rent konkret kan ett rums olika lager utgöras av samma levelement. Vad som skiljer dem åt är spelarens förutsättningar eller angreppssätt.

Backtracking: För att maximalt utnyttja olika lager i en leveledesign, krävs ett visst mått av s.k. backtracking, det vill säga att spelaren måste återvända till samma rum senare i spelet. Leveledesignen kan alltså inte vara helt linjär, utan måste ge spelaren möjlighet att fritt välja vilken väg hon väljer att utforska, samt möjligheten att komma tillbaka till tidigare besökta platser. Framför allt syftar backtracking till att spelaren kommer tillbaka till samma rum med nya förutsättningar. Dessa kan vara positionella, om spelaren t ex stiger in i ett rum från en annan dörr kan det radikalt förändra rummets möjligheter. Det kan också vara så att spelarens avatar har förändrats eller blivit uppgraderad, så att hon nu har möjlighet att interagera med ett annat lager i rummet.

2.2.2 Planerad sequence breaking

För mitt verk så ämnar jag att implementera en stor grad av frihet för spelaren, och använda mig av planerad sequence breaking i leveldesignen. För att kvalificera som sequence breaking och inte klassas som parallella/multilinjära rutter, så vill jag att spelet och leveldesignen implicit ska berätta för spelaren vilken rutt som är den tänkta. Motsvarande vill jag att den sekvensbrytande ruten ska ge spelaren en känsla av att ha hittat något väldigt hemligt, eller kanske till och med ge en känsla av att ha "besegrat" spelet, i det att hon tror att hon hittat något som inte är planerat i designen.

Märk väl att även då mitt främsta mål med leveldesignen är att skapa en kompakthet och en frihet till sequence breaking, så har jag även mer konventionella riktlinjer jag designar efter. Dessa inkluderar "vanliga" alternativa rutter med. Mer om detta i avsnittet "[5.1 Motsättningar mot konventionell leveldesign](#)".

3 Arbetsprocess & Metod

I detta kapitel beskrivs hur jag och min samarbetspartner Daniel planerade upp vårt arbete, min arbetsmetod, och några utvalda designbeslut från dagboken som jag i efterhand också kommenterat.

3.1 Arbetsplanering

Arbetet med verket var indelat i 4 faser. De två sista faserna gjordes om iterativt tills godkänt resultat uppnåddes.

Koncept- & planeringsfas

- Skapa ett antal koncept
- Välja koncept utifrån
 - Genomförbarhet
 - Möjlighet till kompakt leveledesign
- Förutspå och överkomma produktionssvårigheter
 - Versionshantering
 - Level editor
- Planera implementationsordning

Undersökningsfas

- Vilka texter och teorier finns det på området?
- Vad går att applicera på mitt område?
- Finns begreppet ”kompakt leveledesign”?
- Vad är Sequence Breaking?
- Vilka riktlinjer ska vägleda mig?

Designfas

- Skapa ett pusselbibliotek
- Övergripande struktur
 - ”Pacing”
 - Svårighetsgrad
 - ”Sequence Breaking”
- Binda allting samman till något enhetligt

Implementationsfas

- Överföring av design från papper eller andra verktyg till Spelet.
- Buggtestning
- Speltester på testgrupp

- Eventuell iterativ omdesign

I undersökningsfasen letade jag efter tidigare litteratur i ämnet, och läste böckerna *Level Design for Games: Creating Compelling Game Experiences* (Co, 2006) och *Game Level Design* (Byrne, 2004). Ingen av dessa anknöt direkt till kompakt leveledesign, eller något som skulle kunna kallas kompakt leveledesign, så därför har jag valt att inte bygga mina begrepp på tidigare teorier eller texter, utan mina begrepp ”står på egna ben”, och kan ses som en grund till vidare diskussion eller forskning inom ämnet.

Planeringen följdes till stor del. Framåt slutet av arbetet med verket så genomförde jag många speltester och gjorde ändringar efter den återkoppling jag fick av dem. Då gick jag inte nödvändigtvis igenom de sista två punkterna steg för steg, utan gjorde ofta ändringar direkt i Game Maker. Anledningen var att jag av återkopplingen ofta omedelbart fick nya idéer till lösningar på problem som uppenbarat sig, och för att inte hämma kreativiteten, eller i värsta fall glömma min tanke, så ”dök” jag genast in i uppgiften, för att så snabbt som möjligt kunna testa och utvärdera den nya versionen.

3.2 Arbetsmetod

Som nämnt använde jag mig av min definition av kompakt leveledesign under arbetet med verket. Definitionen fungerade som ett sätt att stämma av att mitt arbete höll fokus genom hela arbetet.

För att arbeta fram leveledesignen använde jag mig dessutom av ett pusselbibliotek, samt två hjälpdokument, ett s.k. pacingschema, och en abstrakt rumslayout. Pacingschemat (se Bilaga 2) består av ett antal parallella tidslinjer, en för varje kategori av objekt, pussel, eller övrigt motstånd. Här sätter jag ut vilken ordning allting i spelet ska introduceras i, och i vilken takt. Jag ser till att det sker så lite överlappning som möjligt, och att progressionen är logisk, med tanke på vad spelaren har för föremål att tillgå vid varje givet tillfälle, och vilket motstånd eller pussel det finns vid samma tillfälle. Pacingschemat låg hela tiden som grund till det arbete jag sedan gjorde i rumslayouten. Jag gjorde dock inte klart hela pacingschemat innan jag påbörjade layouten (eller ens implementation/speltestning), eftersom jag gjorde spelet i bitar, ofta plan för plan, och såg till att spelet var spelbart kontinuerligt.

Rumslayouten (se Bilaga 3) beskriver innehållet i varje rum i abstrakta termer. Till exempel så betyder ett område inringat med en lila linje att Teleporteringsstaven krävs för att komma förbi det hindret, men det beskriver inte exakt hur. Rumslayouten beskriver alltså kopplingarna mellan rummen, vad man hittar för föremål i dem, vilka monster det finns där, och vad man behöver för att lösa pusslen. Själva implementationen planerar jag inte i förväg.

Sista steget är implementation av leveledesignen i utvecklingsverktyget Game Maker. Jag följer här rumslayouten så noga jag kan, framför allt vad gäller rummens kopplingar med varann (dörrar mellan rummen, och vad man behöver för att öppna desamma) och var skattkistor med nya föremål finns. För implementationen av pusslen hämtar jag också idéer från ett bibliotek av isolerade pussel som jag skapade i ett annat verktyg i början av arbetet, under tiden jag väntade på att Daniel kunde ge mig en version av spelet att börja arbeta med.

Många misstag i designen kommer fram kontinuerligt, och för att den övergripande strukturen inte ska bli lidande, eller få logiska luckor, går jag tillbaka till rumslayouten och reviderar, istället för att revidera direkt i Game Maker. I rumslayouten har jag nämligen mycket bättre översikt över strukturen, och det går mycket snabbare att ändra i detta dokument än i Game Maker. Jag fick dock ofta brottas med min disciplin för att inte genast börja arbeta på ändringar direkt i implementationen.

Jag är mycket nöjd med detta upplägg av arbetssätt, och hade jag gjort om hela projektet idag hade jag använt mig av samma metod. Det bör dock tilläggas att det är specialutformat för spelet Ittle Dew, och kan inte nödvändigtvis användas omodifierat för utvecklandet av spel i andra genrer, även om jag tror att mycket går att återanvända.

3.3 Utdrag från projektdagboken

Under arbetets gång förde jag dagbok över dels vad jag specifikt jobbade på den dagen, men framför allt vilka problem jag och min arbetspartner Daniel stötte på, hur vi löste dem, och varför vi kom fram till den lösningen. Här följer några valda utdrag ur denna dagbok som reflekterar olika beslut och problem jag stötte på under arbetet, samt en tillhörande kommentar skriven i efterhand.

Det bör nämnas att veckoangivelserna är räknade från projektstart, och är inte årets veckor.

3.3.1 Förproduktion

Vecka 1

Under första veckan bestämde vi oss för att tänka om en aningen från vårt första upplägg av examensarbetet. Vi gick tillbaka till ritbordet och började spåna nya spelkoncept helt fritt. Första kvällen kom det så många idéer att vi märkte att det var alldeles för mycket för ett och samma spel. Det blev också en aning virrigt när vi diskuterade, därför att det var inte helt tydligt vilken uppsättning olika idéer som den ena pratade om. Efter att ha spånat idé A och B, kommer den andra personen med idé C, men menar då att den är istället för t ex B. Sen kommer idéerna D till Ö, och helt plötsligt är det väldigt svårt att hålla alla bollar i luften. Därför valde vi att fullfölja varje stark idé för slutet, och vara ”trogen” den genom att inte ta in för mycket annat som inte passar ihop. Detta ledde till att vi bestämde oss för att göra ett koncept om dagen hela veckan, och sedan utvärdera vilken vi skulle använda oss utav innan måndagen vecka 2. Konzepten vi skrev ihop var:

- Block med förenade attribut
- Zelda-klon med portalgun
- Stackars Städtanten – ett action/pussel-spel med två distinkta faser
- Tetris-labyrint

På fredagen fick vi inte ihop något nytt koncept, för vi var så fästa vid Zelda-klonen och Stackars Städtanten. Daniel var mest fäst vid Städtanten, och jag var mest fäst vid Zelda-spelet. Vi lät idéerna sjuda över helgen för att välja på måndagen.

Vecka 2, Måndag

Efter att ha låtit koncepten från vecka 1 sjunka in, så bestämde vi oss tillslut för att köra på konceptet vi kallade ”Zelda-klonen”. Det stod mellan det och ”Stackars Städtanten”, och vad som slutligen gjorde att vi valde bort Stackars Städtanten var att det saknade den detaljerade designen som gjorde pussel-fasen pusslig och rolig. Konceptet bestod i att det först skulle vara action och sedan pussel som berodde på hur action-fasen utspelat sig, men mer konkret än så blev aldrig designen, fastän vi väntade en hel dag på en snilleblix.

Kommentar

I efterhand är jag glad att vi valde det koncept vi gjorde, men vill fortfarande gärna utforska konceptet ”Stackars Städtanten”. Ittle Dew är dock perfekt lämpad för kompakt leveldesign, och tekniskt sett passade det vårt arbetssätt och projektomfång väl. Se Bilaga 4 för spelets samlade designdokument.

3.3.2 Tidig speltestning

Vecka 6, Tisdag

Jag har bett Daniel att skapa en version av spelet jag kan börja använda nu, även om många saker fortfarande inte är implementerade. Detta ville jag för att jag så tidigt som möjligt ska kunna testa att min design fungerar. I nästan alla projektutvärderingar (post mortems, intervjuer) jag har läst, har de attribuerat ett lyckat spel till mycket och tidig testning. Daniel kommer förmodligen kunna ge mig en version innan dagen är slut.

Vecka 6, Fredag

Jag har genomfört ett antal speltestningar med olika personer och kompilerat en lista med punkter av feedback. Det har gett ganska bra feedback redan, och jag ska inte göra misstaget att hålla på testversioner tills jag känner att de är bra nog att testas. Även om jag själv redan vet många brister med testversionen skadar det inte att få ut den bland testarna och om inte annat, få sina egna misstankar bekräftade. Spelet är dock väldigt spelbart just nu, och det är faktiskt min leveldesign som ligger efter för tillfället.

Kommentar

Som jag skriver ovan och går in på mer i avsnittet [5.5 Vikten av tidig speltestning](#) så är den generella konsensus att tidig speltestning är A och O för utvecklandet av välgjorda och framgångsrika spel. Detta är något jag själv erfarit under tidigare projekt och något jag arbetar noga efter numera.

3.3.3 Nytt designbeslut: Varje våning har unikt utseende

Vecka 6, Söndag

Jag har beslutat mig för att likna varje plan vid en lätt igenkännlig skepnad (första planet ser nu ut som ett paraply, se Ill. 1). Detta för att Spelaren lättare ska känna igen sig och kunna differentiera planen från varandra. Ett plan som använder alla 25 rum ser alldeles för kaotiskt ut på minimappen.

Dessutom kan spelaren försöka gissa sig till skepnaden innan hon får planets karta, och därmed få ett bättre hum om vilka rum som är värda att försöka ta sig till och vilka som är tomma.

Jag ska experimentera med att skapa skepnaden först, och sen se om jag kan passa in en leveldesign i den. Detta får jag dock vara försiktig med, jag vill inte att några andra riktlinjer ska hamna i skymundan på grund av detta. Men om något var det bara en fördel för layoutens kompakthet att jag komprimerade första planet att få plats inuti skepnaden av ett paraply.

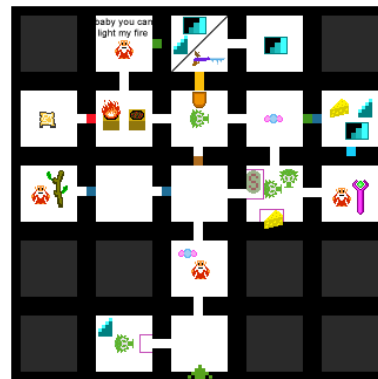


Illustration 1 - Planet bildar skepnaden av ett paraply

Vecka 7, Torsdag

Jag har börjat på källarplanet (plan 3 av 4) idag. Plan 2 är inte klart ännu, men har fått sin övergripande struktur och de flesta av sina pussel.

Försökte hitta en skepnad som passade källarplanet (vissa rum är fasta eftersom det leder ner trappor från planet ovanför), men hittade ingen klockren. Har därför börjat designa rum för rum, och kommer leta efter en skepnad lite senare. Detta kommer kanske att innebära att jag får komprimera rummen i slutskedet och lägga ihop funktioner i samma rum för att spara in plats. Men precis som med första planet kan detta mycket väl leda till en bättre leveldesign.

Kommentar

Jag står fast vid att det var ett bra beslut att låsa varje plans layout till en viss skepnad, dels för att det underlättar för spelaren i det att hon lättare kan komma ihåg skillnaden mellan de olika planen och dels för det tvingar mig att komprimera mina idéer så att det får plats på mindre utrymme än tidigare tänkt. Jag är övertygad om att begränsningar är de mest kreativa förutsättningarna; när man har full frihet att skapa blir man ofta paralyserad i sitt skapande just eftersom möjligheterna är så stora. Begränsningar tvingar skaparen att "tänka utanför lådan".

4 Presentation av verket

Här följer en beskrivning av spelet Ittle Dew, och tre exempel ur spelet som visar på kompakthet, sequence breaks, och även konventionell leveledesign. För en komplett körbar version av spelet, se Bilaga 1

4.1 *Ramverket Ittle Dew*

Ramverket i vilket mitt kärnverk, leveledesignen, är implementerad i är spelet Ittle Dew, ett äventyrsspel till PC utvecklat tillsammans med min arbetspartner Daniel Remar.

4.1.1 Historien

Ittle Dew är namnet på huvudkaraktären såväl som på spelet. Undertiteln är ”- And the lost sidekick”, och storyn går ut på att Ittle och hennes hjälpreda Tipsie är på väg hem från deras senaste äventyr när helt plötsligt Tipsie försvinner. Eftersom Ittle är beroende av Tipsie för att lösa svåra pussel blir hon mycket upprörd och ger sig ut för att leta upp honom. Hon kommer till ett misstänkt slott, och går in för att leta. Här börjar spelet, och hela spelet utspelar sig inuti detta slott, som är fullt med monster och pussel. När Ittle letat igenom hela slottet och dödat varenda varelse däri, inser hon att Tipsie inte är där. Här slutar spelet och visar den sista biten av historien: när Ittle kommer tillbaka till lägerplatsen ligger Tipsie där och sover. Han hade bara gått iväg för att kissa.

4.1.2 Spelmekniken

Spelarvyn är ovanifrån, och visar ett rum i taget. Alla rum är lika stora, och varje våningsplan i slottet består av 5*5 rum. Det finns 4 plan i slottet.

Spelaren styr avataren Ittle, som kan gå i 8 olika riktningar, använda sina 4 olika föremål (om hon erhållit dem, hon börjar tomhänt) samt interagera med diverse objekt så som att putta block och ställa sig på tryckplattor. En tryckplatta öppnar ofta en dörr i rummet, som möjliggör vidare utforskning av slottet, men det finns även många andra typer av pussel.

De fyra föremålen är Eldvapen (i tre olika uppgraderingar), Bomb, Fryspistol och Teleporteringsstav (i två olika uppgraderingar). Alla dessa fungerar i kombination med de övriga tre, t ex kan man frysa bomben så att dess detonation blir fördröjd (samt glider längre när man puttar den), man kan teleportera bomben, frysa en vägg med fryspistolen och studsa en teleportboll på väggen för att i sin tur kunna teleportera iväg en eldboll, etc. Dessutom finns det många olika objekt att interagera med (och många av dem kräver ett visst föremål), så det finns ett enormt antal pussel man kan skapa med dessa relativt få element. Detta knyter an till temat kompakthet som genomsyrat hela projektet.

4.2 *Exempel på kompakt leveledesign*

I illustration 2 nedan visas 4 sammanhängande rum. I en normal spelgenomgång kommer spelaren att stiga in vid (1). Hennes slutgiltiga mål är att ta sig vidare ut vid (2). För att

komma dit måste hon färdas fram och tillbaka flera gånger genom de fyra rummen och låsa upp alla de pusslen som finns i dem.

Som bakgrund till hur pusslen fungerar i Ittle Dew tänker jag här förklara en detalj om öppna och stängda dörrar. En dörr existerar alltid i två rum, och låser du upp en dörr, kommer den alltså vara öppen från båda rummen. En öppnad dörr förblir dessutom också för alltid öppen. När spelaren alltså vid (3) utlöser tryckplattan och därmed öppnar dörren vid (4), så öppnar hon alltså snarare upp dörren i det nordvästra rummet än det sydvästra som hon är i (hon låser upp båda, men den intressanta är i det nordvästra), eftersom hon så småningom får gå tillbaka och komma via dörr (5) för att tillslut gå igenom dörr (4). Detta är bara ett exempel på hur man som spelare får gå fram och tillbaka för att lösa hela det pusslet som visas här nedan; totalt finns här inte mindre än 7 pussel. Hur dessa fyra rum hänger samman är ett exempel på kompakt leveldesign. Det kan dessutom tilläggas att det är fullt möjligt att komma från (2) första gången man kommer till detta ställe i slottet, varpå förutsättningarna förändras helt.



Illustration 2 - exempel på kompakt leveldesign

4.3 Exempel på planerad sequence breaking

I illustration 3 nedan visas det första exemplet på planerad sequence breaking (SB) i spelet. Denna SB möjliggör en tidig tillgång till både Bomb och Fryspistol, och hoppar helt över Teleporteringsstaven, som i normala fall är det andra föremålet man hittar.

Elden vid (1) kan användas för att tända fyr på sin pinne, och i tur tända eld på facklan mittemot. Detta öppnar dörren till vänster i det rummet. Men spelaren kan här istället använda elden till att tända fyr på den gamla mannen (2). Detta är mycket svårt att utföra på grund av att elden bara är tänd på pinnen i en viss tid, och sträckan till den gamla mannen är precis så lång så att man klarar det, med 2/30 sekund till godo. Om spelaren lyckas så öppnas dörren vid (3), vilket gör att hon kan gå en trappa upp (4) och där uppe hämta bomben. Normalt sett måste hon gå en helt annan, mycket längre väg, för att hämta bomben. Oavsett vilken väg spelaren hämtar bomben kommer hon dock efteråt ta trappan ner till (4), men beroende på om hon använt SB eller inte kommer dörren vid (3) vara låst eller öppen. Är den öppen, som i vårt exempel, kan hon lätt gå runt hela vägen till (5) för att där spränga upp en öppning (se sprickan i väggen som indikerar att man kan bomba sönder väggen där) till föremålet i kistan vid (6), Fryspistolen. Om hon däremot tar bomben via den normala rutten, kan hon inte direkt nå (6), utan måste ta en trappa ner, gå igenom hela det planet, för att så småningom komma upp igen, och komma in i rummet vid (7), för att slutligen få fryspistolen vid (6).

Denna SB skalar uppskattningsvis av 30 min på speltiden; som jämförelse tar en normal spelgenomgång ca 1h – 1h 20minuter (inga SB's använda) och mitt personliga rekord ligger på under 6 minuter (alla SB's använda).



Illustration 3 - Exempel på planerad sequence breaking

4.4 Exempel på konventionell leveledesign

Som motvikt till mitt arbetes, och framför allt denna rapport, fokus på kompakthet, tankar jag här ta upp ett exempel i spelet där jag valde att frångå mina egna mål och välja en mer traditionell design (se vidare definition i avsnittet [5.1 Motsättningar mot konventionell leveledesign](#)), och varför.

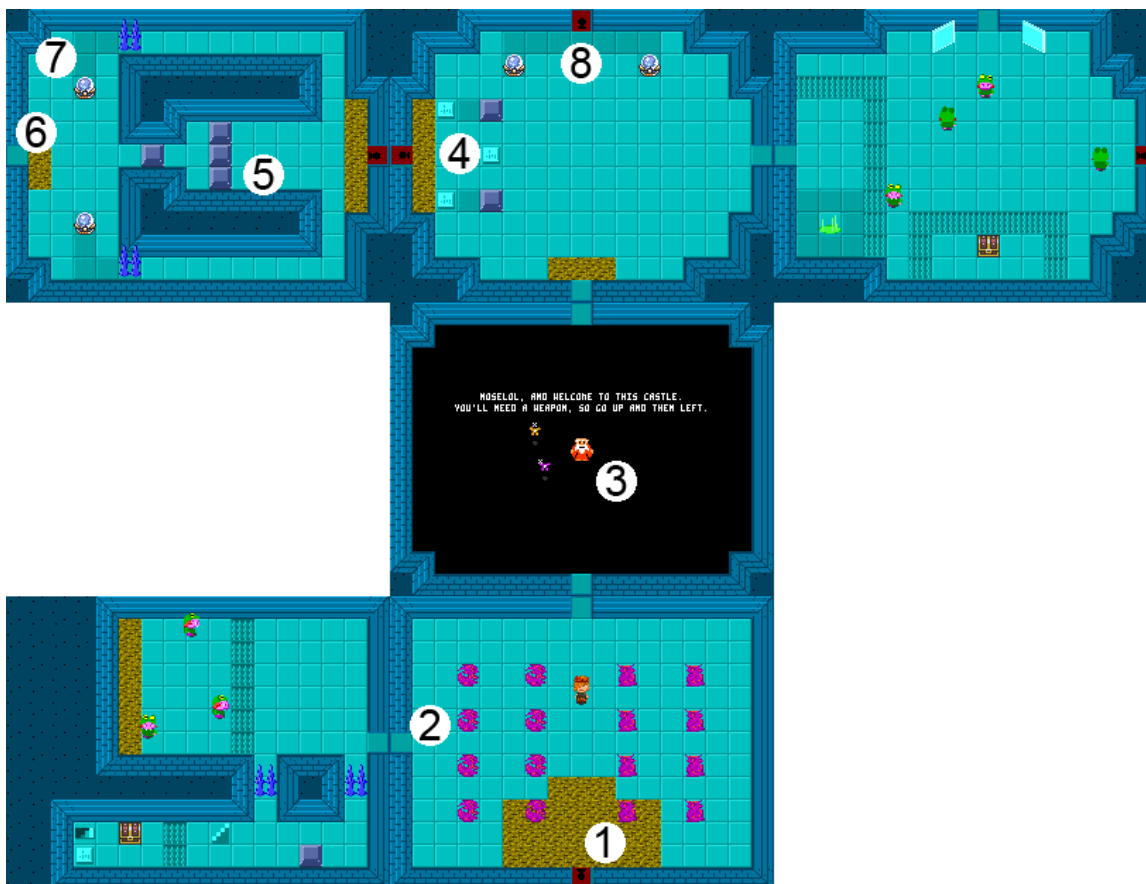


Illustration 4 - Spelets första rum

Illustration 4 ovan visar spelets allra första början (1), och de följande rummen. Ett designval jag vändades mycket över var om jag skulle ha två vägar från första rummet eller inte. Dörren vid (2) leder in till ett rum man inte kan komma vidare i förrän en bra bit senare i spelet (detta är ett exempel på backtracking), och min förhoppning är att spelaren ska inse detta om hon väljer att gå dit det första hon gör. Speltesterna har visat att så är fallet också. Det potentiella problemet med att ge val så tidigt i spelet är dock att spelaren riskeras att överbelastas med information, och jag funderade länge på om jag skulle ha det så eller inte. Här fick dock den konventionella leveledesignen stryka på foten för min vision, speciellt med tanke på att det inte visade sig vara något problem i speltesterna. Utöver detta exempel har jag dock hållt inledningen relativt linjär, och försökt att introducera en sak i taget, så att spelaren ska lära sig spelet i en lättsmält takt. Nästa rum (3) visar en gammal man som ger spelaren motivation i form av ett hägrande

vapen och var det finns (i rummet genom (6)). Därefter introduceras det första pusslet (4), vilket går ut på att med hjälp av två puttbara block, trycka ner tre tryckkänsliga plattor samtidigt (spelaravataren trycker själv ner den tredje) för att öppna dörren till vänster. Eftersom spelaren har fått en hänvisning om att gå till vänster, så behöver hon ännu inte bry sig om att interagera med pusslet i norra delen av rummet, eller att gå genom dörren till höger. Försöker hon trots allt med något att detta kommer hon inte komma långt, och kommer inom kort ge sig på pusslet till vänster för att ta sig vidare. Nästa pussel (5) är ytterligare ett som involverar att putta block, men problemet består i att ta sig förbi dem. Detta pusslet har visat sig ge huvudbry för ett litet antal av testpersonerna, men de allra flesta lyckas lösa det inom 1 eller 2 minuter. Nästan alla testpersoner har dock misslyckats på första försöket, varpå en annan egenhet av spelet introduceras; pussel återställs om man antingen håller inne en knapp i ca 5 sekunder (detta står tydligt på skärmen genom hela spelet), eller om man går ut och in i rummet. Alla testpersoner har tidigt insett detta, vilket med största sannolikhet har att göra med att de är vana vid att spel oftast fungerar på detta sätt. I rummet genom dörren vid (6) får spelaren sitt första föremål, ett vapen i form av en pinne. För att sedan ta sig ut igen vid (7) måste hon använda pinnen på kristallen för att sänka spikarna och ta sig ut (pusslet vid (5) går inte att lösa från det hållet), och kommer nu snabbt inse hur man även löser kristallpusslet vid (8) för att komma vidare till rummet mot norr.

Om mitt enda mål med spelet hade varit att ha så stor kompakthet och tillåta så många sequence breaks som möjligt, hade inledningen sett helt annorlunda ut, med mängder av val och obegripliga pussel som kan lösas på mer än ett sätt för att öppna hemliga vägar. Men på det stora hela är alltså inledningssekvensen väldigt linjär, och introducerar en sak i taget så att spelaren inte blir frustrerad och slutar spela. Det är dock redan såhär tidigt i spelet ofta mer än en sak att göra vid varje punkt; man kan undersöka ett "sidorum" eller försöka interagera med ett pussel som inte är tänkt att lösas vid det laget. Detta visar på spelets kärna, kompakthet, utan att vara påträngande eller förvirrande. Att detta verkligen är fallet visar speltesterna på, och är också ett resultat av en stor mängd omarbetningar av designen, dels efter eget huvud, men framför allt efter återkoppling från speltester.

5 Diskussion

Här följer tankar och slutsatser som jag har kunnat dra, både specifikt om mitt eget begrepp ”kompakt leveledesign”, och leveledesign i övrigt och hur det passar in i den mer övergripande speldesignen.

5.1 Motsättningar mot konventionell leveledesign

Det finns många riktlinjer att använda sig av när man designar ett spel eller mer specifikt designar en level till ett spel. Byrne skriver såhär i sin bok *Game Level Design* från 2005 (min översättning):

... det finns universella koncept inom leveledesign som du kan lära dig att identifiera och manipulera för skapa roliga upplevelser ... I ett nötskal är dessa Ergonomi, Flöde (Flow, psykologisk term), Rytm, Svårighetsgrad, Wow-faktor & ”Hooks” (speciella egenskaper att hänga upp sitt verk på)

Många av dessa riktlinjer går tvärt emot både kompakt leveledesign och framför allt planerad sequence breaking. Du bör t ex som leveledesigner undvika att i början av spelet introducera för mycket val, för många olika element eller för många spelregler. Spelaren bör helst lära sig en sak i taget. Å andra sidan bör man i ett spel med väl planerad sequence breaking mycket tidigt kunna hoppa av från den tänkta rutten och skippa stora delar av spelet. Om man som spelare från början är väldigt låst till en rutt som sträcker sig allt för långt in i spelet, börjar hela idén med planerad sequence breaking förlora sin mening.

För att tackla detta söker jag finna sätt att tillåta sequence breaking tidigt utan att belamra nybörjarspelaren med för mycket information. Här följer ett exempel som återfinns mycket tidigt i spelet (notera dock att i slutversionen av spelet ser pusslet lite annorlunda ut, men tanken är densamma).

Spelaren stiger in i ett rum från öster. Rummet innehåller två facklor, varav en är tänd. Till väster finns en låst dörr, och i norr finns en öppen dörr. I rummet mot norr finns en gammal man som säger ”Baby you can light my fire”. Detta syftar på de tidigare rummet, där spelaren ska tända eld på sin pinne med den tända facklan, och med pinnen tända den släckta facklan. Då öppnas dörren till väster. Detta är den tänkta huvudrutten men här finns dock också en alternativ rutt. Om spelaren tänder sin pinne på facklan och går tillbaka till den gamla mannen i det norra rummet, kan hon tända eld på mannen istället, och därmed låsa upp en hemlig väg i det rummet.



Illustration 5 - Dubbeltydigt pussel med två lösningar

Min förhoppning är att ovanstående sekvens inte ska förvirra en nybörjarspelare med för mycket information. Om hon lyckas tända eld på den andra facklan och klara det pusslet,

är min förhoppning att hon ska se pusslet och ledtråden som löst, och därmed inte tro att det finns mer att göra. Att hela tiden ha i bakhuvudet att det finns mer att göra (därför att spelet visar för mycket av sig själv för tidigt) kan leda till överbelastning och förvirring hos spelaren. Samtidigt vill jag att den avancerade spelaren ska kunna få något ut utav de tidiga delarna av spelet även när hon spelar om spelet en andra gång, och kunna utforska nya vägar redan från start.

5.2 Leveldesign är inget avskärmat arbetsområde

Ju mer jag arbetar och tänker på det arbete som hamnar under området ”level design”, desto mer övertygad blir jag om att detta arbete (beroende på genre) är det mest centrala inom speldesign, och själva implementationen av alla de idéer man har beskrivit i designdokumentet. Det är också här man märker hur väl eller dåligt en idé fungerar, här man sätter stämningen, och även här man till viss del berättar historien.

Naturligtvis finns det exempel på arbetsuppgifter inom leveldesign som är separata från andra områden, men då främst inom mycket stora utvecklingsgrupper där antalet projektmedlemmar uppnår 100-talet och behovet av projektledning är stort och mycket specifika arbetsuppgifter tilldelas medlemmarna. Inom projektet *Little Dew* utgjorde jag 50% av utvecklingsgruppen, och mina ansvarsområden svämmade så småningom ut över alla områden som inte rörde programmering och tekniska detaljer. Av naturliga orsaker blev det efterhand så att det var främst jag som ansvarade och kämpade för designen av spelet, medan min samarbetspartners perspektiv främst låg i den tekniska implementationen och vägde varje designdetalj mot hur svår implementationen av den var i utvecklingsverktyget. Det var under dessa förhållandena som jag som leveldesigner men i förlängningen även som speldesigner, författare, ansvarig för pacing, struktur, etc., kände hur centralt arbetet med leveldesign är. Sådär uttrycker Ed Byrne det i hans bok *Game Level Design* från 2004:

A level designer is the point of convergence for programming, cinematography, audio, art, and design - all of the components of a modern computer or video game...

5.3 Vikten av tidig konkret leveldesign

Jag ska även ta tillfället i akt och dela med mig av en erfarenhet från ett tidigare projekt, spelet *The Blockers*, i vilket vi inte hade någon dedikerad leveldesigner utan delade den uppgiften mellan oss och så att säga ”klämde in” det mellan andra arbetsuppgifter.

I det inledande arbetet med designdokumentet så beskrev vi nästan uteslutande spelmekniken i spelet. *The Blockers* är ett kompetitivt pusselspel där två spelare pusslade och kämpade mot varann samtidigt, för att nå först i mål. Designdokumentet beskrev alla de pusselement som fanns i spelet, hur avatarerna styrdes och vilka förmågor de hade, olika spellägen, och kriterier för vinst. Ingen stans i dokumentet beskrevs ett konkret pussel med illustrationer eller förklaringar, utan den enda informationen till hur de olika banorna kunde se ut, fanns mellan raderna i de detaljerade beskrivningarna av pusselementen. Leveldesignen var något som vi lämnade därhän, därför att vi ansåg att det som bar spelet var den snillrika spelmekniken, och att leveldesignen skulle lösa sig själv. Med spelmekniken i hamn, trodde vi att spelet

automatiskt skulle bli bra och roligt. Det kan tilläggas att vi inte explicit avstod från att beskriva konkreta pussel eller leveledesign, utan anledningen till att den uteblev från designdokumentet var att vi inte nämnvärt reflekterade över den.

Vid projekts slut så hade vi ett komplett spelbart spel, men spelet var inte särskilt kul, och levde inte upp till spelets s.k. high concept (fritt översatt ”övergripande designmål”), vilket löd ”multiplayer puzzle mayhem”. Anledningen till att det föll kort från det målet var inte en brist i spelmekaniken, utan att leveledesignen var ogenomtänkt och inte tillräckligt integrerad med resten av designutvecklingen. Endast två av de 50-talet rum vi skapade tillät spel på den höga nivå som vårt high concept beskrev.

Hade vi istället fokuserat på att mycket tidigt utvecklat prototyper av konkreta banor som höll måttet hade vi förhoppningsvis kunnat förändra spelmekaniken så att den passade bättre till de leveledesigner vi kunde producera, eller i värsta fall kommit fram till att spelet behövde ett helt ny inriktning eller kanske helt läggas ner.

5.4 Samarbetet – intressekonflikter och kompromisser

Efter det att jag och Daniel hade bestämt oss för ett koncept, skrivit designdokument till det och satt igång med arbetet, började så småningom nya roller utkristallisera sig som inte var planerade. Eftersom Daniel tampades med programmering och allt vad det innebär, var hans främsta angelägenhet (detta har bekräftats av honom via dialog i efterhand) hur lätt eller svår en ny idé var att implementera, och uttryckte ofta motvillighet när jag plötsligt ville få med ytterligare en sak i spelet.

Under konceptfasen beaktades bådas idéer lika mycket, och vi delade lika börda på designarbetet. Men allt eftersom produktionen led längre så var det nästan enbart jag som stod för ”designen”, det vill säga att komma på nya idéer, att kritiskt utvärdera de idéer vi redan hade, och stå för den övergripande kvalitetskontrollen.

Det var naturligtvis mycket lätt för mig att presentera en ny idé till spelet, när det sällan innebar merarbete för mig, och det är fullt förståeligt att Daniel ofta hade en skeptisk inställning till nya idéer, då det alltid innebar merarbete för honom. I slutändan lyckades vi dock kompromissa fram idéer som vi båda är nöjda med, och de allra flesta nya idéer jag bad om visade sig ta mindre än en timme var att implementera. Att de tog så kort tid berodde en del på tur (godtyckliga val från Daniels sida under implementation av X visade sig passa utmärkt med Y), samt att vi arbetade i Game Maker, ett utvecklingsverktyg som är mycket snabbt att arbeta i.

5.5 Vikten av tidig speltestning

Det är svårt att dra några slutsatser om vikten av tidig speltestning kontra sen speltestning utifrån ett enda projekt, men både *The Indie Game Development Survival Guide* (Michael, 2003) samt flertalet av de post mortems man finner efter en sökning på internetsidan Gamasutra (www.gamasutra.com) på sökorden ”early testing” eller ”test early” hävdar att tidig testning har varit centralt för framgången av spelet i fråga. Personligen är jag också fullt övertygad om att många av de problem som jag i detta projekt tidigt stötte på, och rättade, hade skapat enorma problem om de upptäckts mycket senare i produktionen, när betydligt mer spelinnehåll *efter* problemet blir påverkat av ändringar och får göras om. Redan i detta fall blev implikationerna av strukturella

ändringar betydelsefulla, i vissa fall rörde det sig om ändringar som gav upphov till merarbete som sträckte sig över en hel dag. Dessa strukturella ändringar gjordes i min s.k. rumslayout (se [3.2 Arbetsmetod](#)), och inte i själva implementationen, vilket innebar en oerhört mycket mindre arbetsbörda än om jag skulle vara tvungen att göra om även implementationen, och likväl var merarbetet stort. Därför drar jag slutsatsen att tidig speltestning var fundamental för genomförandet av detta spelprojekt.

5.6 Problem med pacing och spelomfång

Ett problem jag har upptäckt efter det att spelet blivit helt spelbart från början till slut och efter att ha genomfört speltester, är att spelaren ofta inte riktigt har tid att "smälta" det hon just lärt sig, eller vad det nya föremål hon just hittat öppnar för nya möjligheter.

Om jag jämför Ittle Dew med sina inspirationskällor är den största skillnaden omfånget på spelet. En spelgenomgång tar i snitt 1h 20min för förstagångsspelaren, att jämföra mot zelda- och metroidspelen som brukar ta 10-15 timmar, om inte mer. I de spelen får man spendera mycket mer tid med varje ny del av spelmekaniken innan spelet presenterar en ny del av sig själv, än vad man får i Ittle Dew. Detta är ett problem när många av pusslen i Ittle Dew består av att kombinera olika saker spelaren har fått lära sig i samma pussel. På detta sätt blir pusslen mer och mer avancerade ju längre spelet lider.

Jag märkte under vissa speltester att testspelaren var fullständigt ställd inför vissa problem, fastän lösningen på problemet precis hade presenterats i rummet innan. Jag frågade hur spelaren tänkte, och bad henne tala högt medan hon försökte lösa pusslet, och det visade sig ofta att problemet var att hon helt glömt bort ett användningsområde av sina föremål, fastän hon använt det bara ett eller två rum tidigare. Vad som saknades var en längre, isolerad experimentering med det nya föremålet eller användningsområdet, där spelaren får leka fritt med det hon just upptäckt, och kanske möjligtvis också *måste* använda det ett flertal gånger innan hon får tillträde vidare. I Ittle Dew används ofta nya föremål bara en eller två gånger innan man blir tvungen att trappa upp komplexiteten och kombinera det med något gammalt, eller på egen hand upptäcka en "latent" förmåga hon haft hela tiden.

Lösningen på detta designproblem är att göra ett längre spel, eller skära ner på antalet förmågor som spelaren har till hands.

6 Slutsats

Med avstamp i personliga betraktelser jag gjort av spel som inspirerat mig (främst metroidserien), har jag definierat begreppen kompakt leveledesign och planerad sequence breaking, som jag anser finns i dessa spel (men som inte finns definierat i tidigare litteratur). Med denna definition som utgångspunkt har jag implementerat en leveledesign i spelet Ittle Dew som jag samproducerat med Daniel Remar. Återkoppling från åtskilliga speltestare av spelet visar på att det i nuvarande form är uppskattat och håller god kvalitet.

Den höga nöjdheten hos speltestarna attribuerar jag till den solida speldesignen, samt den genomarbetade leveledesignen, men värt att nämna är att jag har varit mycket noga med att inte gå överstyr med min vision i leveledesignen, utan kontrollerat att den håller god kvalitet även efter mer traditionella standarder. Denna självkontroll har varit nödvändig då det annars är lätt att ”bli elak” som leveledesigner och utforma alltför krävande pussel och motstånd som bara frustrerar den normale spelaren.

Kompakt leveledesign är något jag är högst intresserad utav, och något jag gärna ser i fler spel. Jag uppmuntrar de som läser denna rapport att utmana mina definitioner och att de producerar egna spel med liknande teman eller riktlinjer.

Referenser

Böcker

Byrne, E (2004) *Game Level Design*. Boston, Charles River Media

Co, P (2006) *Level Design for Games: Creating Compelling Game Experiences*. Berkeley, New Riders Games

Michael, D (2003) *The Indie Game Development Survival Guide*. Boston, Charles River Media

Spel

Nintendo R&D4. *The Legend of Zelda*. Nintendo. 1987, (Nintendo Entertainment System)

Nintendo EAD. *The Legend of Zelda: A Link to The Past*. Nintendo. 1992, (Super Nintendo Entertainment System)

Nintendo R&D1, Intelligent Systems. *Super Metroid*. Nintendo. 1992, (Super Nintendo Entertainment System)

Retro Studios. *Metroid Prime*. Nintendo. 2002, (Nintendo Gamecube)

Nintendo R&D1. *Metroid*. Nintendo. 1986, (Nintendo Entertainment System)

Nintendo R&D1. *Metroid: Zero Mission*. Nintendo. 2004, (Game Boy Advance)

Bilaga 1 - Spelet Ittle Dew

Se medföljande CD-skiva för åtkomst av bilaga 1.

Bilaga 2 - Pacingschema för Ittle Dew

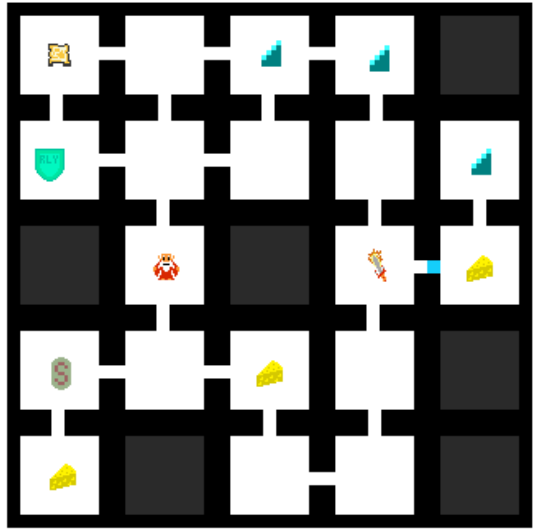
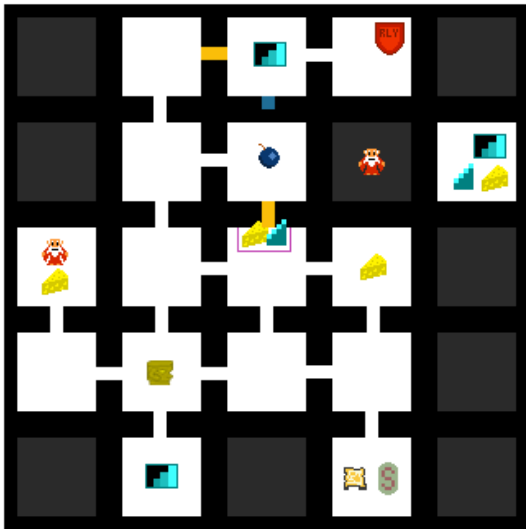
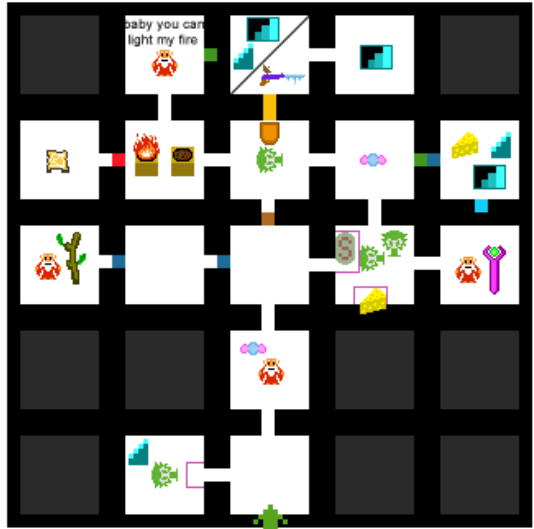
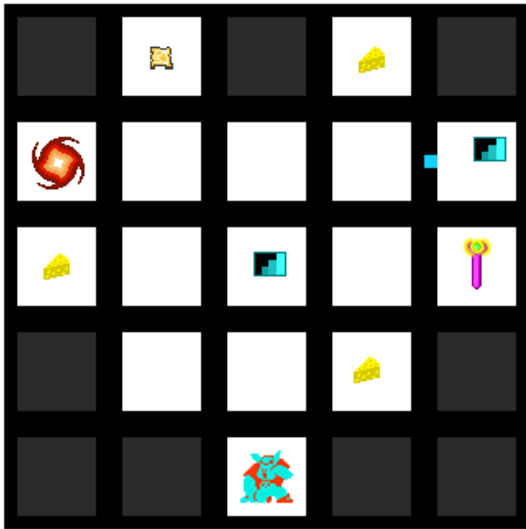
Nedan visas en förhandsgranskning av det Photoshop-dokument som utgör bilaga 2. Se medföljande CD-skiva för full åtkomst av denna bilaga.

	0	10	20	30	40	50	60	70	80	90	100
Tools											
Monsters											
Baguettes											
Puzzles											

Bilaga 3 - Rumslayout för Ittle Dew

Nedan visas en förhandsgranskning av det Photoshop-dokument som utgör bilaga 3. Se medföljande CD-skiva för full åtkomst av denna bilaga.

Open:	■	Pressure:	■	Kill all:	■
Melee:	■	Bomb:	■	Ice:	■
Portal1:	■	Fire:	■		



Bilaga 4 – Designdokument för Ittle Dew

Ittle Dew – Design

Avataren

Gå

Putta

Tryck mot object viss tid för att putta det. Avatar står kvar (om inte fortsätter att trycka). Alla block, isblock, bomber etc som kan knuffas snappar till tiles.

Verktyg:

Portal Gun

Ice wand

Bomb

Melee

Stick

Fire sword

Flame Thrower

Sköld

Framme i normalläge

Försvinner när man använder verktyg

Uppgraderas i steg

Normal

Big – skyddar mot större projektiler

Mirror – Reflekterar projektiler

Kontroller

Styrs med joypad. D-pad + 4 face buttons

Varje verktyg har egen knapp.

Verktygen

Portal Gun

Tryck 1 gång för att skjuta ut destinations-boll. Relativt låg hastighet.

Trycka 2'a gång för att skjuta ut teleport-boll. Denna har högra hastighet. Det objekt den träffar teleporteras till destinationsbollens nuvarande position. Om teleportbollen inte träffar ett relevant objekt så sker ingen teleport. Destinationen försvinner när teleportbollen försvinner.

Båda bollar studsar mot reflekterande ytor.

Uppgraderingar:

Första versionen kan inte teleportera knuffblock/isblock
Andra versionen kan teleportera även knuff-och isblock

Ice Wand

Skjuter ut isboll. Isbollen formas en eller två tiles framför spelaren.
Isboll fryser monster, spelaren, block, bomber. Frysta objekt glider extra långt när de puttats. Frysta objekt töar efter viss tid.
Isboll släcker eldar. Detta kan utlösa en trigger.
Isboll studsar mot reflekterande ytor.
Om en titan (monster som blir block när det fryses) försöker frysas men hamnar på samma tile som avataren, fryses det inte utan isbollen försvinner istället.

Bomb

Oändligt förråd. Bara en kan finnas ute i taget. Bomber läggs ut en tile framför spelaren.
Frysta bomber har avpausad detonering. Vid upptining fortsätter detoneringstiden.
Bomber kan puttats en tile i taget. Frysta bomber fungerar som alla frysta block och glider.

Meleevapen

Stick (pinne)

Slår framåt, kort räckvidd.
Kan slå fiender, triggas av triggers.
Man kan sätta eld på pinnen så får den eldattribut. Slocknar efter ett tag.

Fire sword

Slår framåt, lång räckvidd.
Eldattribut. Smälter is och tänds bomber.
Starkare än pinnen.

Flamethrower

Skjuter eld framåt, en salva i taget. Elden försvinner när den träffar väggar och triggas.
Mycket lång räckvidd. När över håll.
En "bit" av elden kan teleporteras om den träffar en teleportboll.
Starkare än eldsvärdet.
Sätter eld på féer. Röjer fett.

Level objekt

Triggers

Kan:

Öppna en låst dörr
Trolla fram en pinata

Pressure pads – kan triggas av block, frysta stora fiender, bomber.
Kristall – triggas av slag med melee, bombexplosion, ice wand, dest/teleportboll.

Door switch – sitter precis jämte dörren, annars som kristall
Eldar – Släck med ice wand för att trigga
Släck eld – tänd med eld för att trigga
Triggers som är aktiverade förblir aktiverade efter de ”använts” (den öppnade dörren har passerats, den framkallade piñatan är öppnad).

Monster no zone

En tile. Monster går inte hit. Används för att blockera monster.

Block no zone

Ittle no zone

Block kan färdas över dessa

Reflekterande ytor

Reflekterar alla typer av projektiler. utgångsvinkel inverterad mot ingångsvinkel.
Spegel, kan luta 45 grader.
Isblock är reflekterande.
Mirror shield är reflekterande.
Is-vägg – bildas av att skjuta ice wand in i väggen. Is-vägg töar efter ett tag, och förstörs omedelbart av bombexplosioner och eld (väggen blir en vanlig vägg igen).

Block

Kan puttats en tile i taget, endast om det inte finns något i vägen på nästa tile, inklusive fiender. Puttning sker genom att avataren trycker sig mot blocket i en viss tid.
Triggar pressure pads så länge det är direkt över dom.
Block kan frysas med Ice wand och blir då isblock som glider tills det träffar en vägg när det puttats. Töar efter viss tid. Töar även när de träffas av bombexplo / eld.

Isblock

Kan puttats – glider tills det träffar en vägg.
Triggar pressure pads sålänge det är direkt över den.
Töar aldrig på egen hand.
Förstörs av bomber och eld.
Är en reflektor.

Piñatas

Innehåller skatter så som nya verktyg, hälsouppegradering etc.
Öppnas genom att slå med melee
Finns max 1 per rum.
Med hjälp av en trigger kan piñatan göras oanvändbar (den sitter bakom galler eller liknande). Den görs användbar igen med en (eller flera) triggers, dvs aktiveras precis som en dörr.

Dörr

Kan öppnas med triggers av olika slag.

En upplåst dörr förblir upplåst. (När alla tillhörandetriggers är aktiva samtidigt låses dörren upp permanent)

Sjunkande pelare

Kan stå mitt i rummet.
Blockerar vägen.
Sjunker ner i golvet när triggad
Kan inte interageras med i övrigt.

Trappa

Placeras i rummet.
Leder mellan plan.

Old Man Kundvagn / Father Forah

Finns utspridda över hela borgen. Säger kryptiska ledtrådar (en per gubbe) som har att göra med en närliggande hemlighet / pussel.

En special-gubbe finns i ett speciellt rum. Hasplar ur sig fänigheter och halvsanningar på slump när man går in i rummet. Har en stor mängd repliker att slumpa mellan

Exempel: "Use buttons to win the game."

Fiender

1. Faerly Harmless

Undviker pinnen, flame sword, bomber, ice wand, portal gun, men kan träffas.
Kan grillas med flamethrower.

"Hey!" "Watch out!" "Hello!"

Ingen attack.

Rörelsemönster: Flyger runt och är irriterande. Spelar egentligen ingen roll hur den rör sig.

2. Jenny Rich Monsteur

Vanlig mob.

Finns i tre färger: grön, gul, blå.

Grön: Går runt och petar med svaga attacker. Kan blockas med skölden.

Gul: Kastar yxor. Kräver big shield eller bättre.

Blå: Skjuter med laserpistol. Kräver mirror shield. Farliga.

Fryser man den stannar den.

Rörelsemönster: Går mest runt på slump och attackerar när spelaren kommer i skottlinje. Ju lägre rank, desto mer slump, och ju högre rank desto mer jagar den aktivt spelaren.

3. Titan Appde'Grafiques

Tung, hård, fyrkantig form. Bli isblock när de fryses.

Melee attack, kan stoppas med big shield eller bättre.

Rörelsemönster: Hoppa en tile i taget. Rör sig slumpmässigt tills spelaren kommer tillräckligt nära, då blir hon jagad. När den är i luften kan den inte frysas.

4. Noselol / Stack of pancakes

Äter bomber och portaler. Bombarna/portalerna försvinner, fienden lever. Jagar närliggande bomber och portaler.

Fryser man den stannar den.

Fångar (äter) spelaren, man måste slå sig ut så dör den. Ingen sköld hjälper mot att bli uppäten.

5. Turnip

Svänger höger när den träffar en solid vägg eller block. Den börjar alltid med att gå uppåt. Puttar bomber och block framåt en tile och svänger då.

Fryser man den stannar den.

Elektrifierar spelaren om hon kommer för nära (area attack). En tiles räckvidd. Inga sköld hjälper eftersom det är en area attack.

6. Ismonster

Keldas Darknuts. Har reflekterande skydd framåt, massiv sköld.

Svåra att döda. Eld och bomber skadar dom mer. Får tillbaka hälsa av isattacker.

Attackerar med ice breath. När två tiles. Fryser Kelda, man får masha sig ut. Ice breath stoppas med big shield eller bättre.

7. Eldmonster

Man kan tända pinnen på den, men monstret tänder inte eld på saker själv.

Alla meleevapen är ineffektiva. All eld är alltså ineffektiv.

Försvinner när man skjuter den med ice wand.

Bomber blåser ut dom.

Saknar attack, men Kelda skadas av att nudda dom. Kelda får eld i baken och springer runt svårkontrollerad.

Hälsa

Börjar med 1 baguett.

1 baguett kan skadas i 4 steg.

Genom att hitta 4 baguettdelar i piñatas så bildar man en hel ny baguette.

De fyra delarna är grafiskt annorlunda från varann, men man hittar alltid nästa del i turordningen.

Bröd

Ost

Sallad

Tomat/Pastrami/Jordnötssmör/Kolasås – beroende på vilken baguette man samlar

till

Man kan samla ihop till totalt 4 baguetter.

Gränssnitt

HUD visar baguetter och de 4 verktygen, samt minimap

Pausemeny

- Visar alla upgrader
- Full karta
- Load checkpoint
- Exit Game

Startmeny

- Start
 - 1, 2, 3
- Load
 - 1, 2, 3
- Options
 - Sound level
 - Music level
 - Set buttons...
 - Bla
- Exit

Karta

Minikartan fylls i efterhand man kommer in i nya rum och hittar skattkistor.

Alla rum är gråa tills man varit i dem.

Beträdda rum blir vita, och rum-kopplingar och dörrar visas

När man hittar kartan blir obefintliga rum helt svarta, och obeträdda förblir gråa

Kan hitta kartor över hela planet som visar:

Dörrar (om dom ännu inte är permanent öppnade)

Skattkistor

Bombade hål

Verktyg visas med extra viktig ikon.

Finns bara en piñata eller trappa per rum, den visas mitt i rummet.

Kartan i pausemenyn visar alla våningar i mer detalj.

Sparning

Finns utplacerade spar-stationer som måste aktiveras genom att slå på med melee.

När man dör får man valet att spara, avsluta eller fortsätta. Om man väljer fortsatt så

sparas spelet samt man hamnar på den senast aktiverade sparstationen.

När man spawnar så har man kvar alla items + uppgraderingar, men har lika mycket liv som man började med.

Finns 3 sparprofiler

Boss

Vi får se vilken som hinner göras:

1

Lite grafik och lite dialog, men ingen bossfight. Lal!

2

Lite grafik och dialog, man kämpar mot bossen genom val i dialogen.

3

Lite grafik och dialog. Man får spela ett minispel mot bossen, typ Pong.

4

Full-fjädrad boss utan fjädrar. Man måste använda items etc. Bossen har kanske flera steg.

Endgame

Vi får se vilken som hinner implementeras:

1

Cutscene bestående endast av stillbilder och text.

2

Cutscene bestående lite av in-game med bossen, och sedan med stillbilder och text. En resultatskärm där det står tid, pengar, kanske lite stats.

Teknisk bibel

Dimensioner

Spelets upplösning är 640*480 pix

Varje ruta är 32*32 pix

Varje rum består av 17*13 rutor

Aktivt rum är justerat till övre högra hörnet av skärmen (96,0 – 640,416).

Varje plan är 5*5 rum

Det finns 4 plan

Kartutritning

Varje plan är ett separat GM-rum.

Varje plan har ett antal rum. När man är i ett rum är objekt i alla andra rum inaktiva, och de omliggande rummen syns inte (antingen täcker ett rum hela skärmen, eller så blir det en svart kant runt rummet).

Objekten som sätts ut är spawn-objekt som spawnar master-objektet i kod. Spawn-objektet har samma grafik och dimensioner som master-objektet.

Ghost objekt ger även sitt master-objekt sin individuella trigger-siffra och övriga instans-specifika inställningar.

Triggers

För att koppla triggers får instanser av objekt en individuell siffra som hör ihop med ett annat objekts siffra i rummet (andra rums objekt är inaktiva). En trigger skickar iväg en signal till alla objekt av typen dörr (och andra objekt som kan triggas om det finns) och anger siffran. De som har samma siffra anser sig själva som "triggade".

Det objekt som blir triggat håller reda på hur många triggers den har kopplad till sig. Den måste få signal från lika många under samma update för att bli triggad.

Rum

Rum kan bara ha dörrar i mitten av väggen. Detta gäller också för bomb-sprickor (som är en annan typ av dörr).

Intelligande rums gemensamma dörr låses upp för båda rummen så fort den har blivit passerad. tekniskt sett är det en dörr per rum med olika trigger-kopplingar.

Minneshantering

Dörr array

Pinata och trapp array

Monster på ett plan förblir döda så länge man är på planet. Vissa fiendetyper spawnas dock alltid om varje gång man går in i rummet

När man byter plan nollställs alla monster

Manuskript

Crazy Old Man

Use buttons to win the game.

It is in the what not true about bombs.

Titans doesn't like certain kind of Ice Wand.

Faeries are highly flammable. Use with care.

10th enemy set up us the bomb.

What does the scouter say about your money?

(om spelaren har över 9000 pengar)

What nine thousand!?!

Over NINE THOUSAND!

It's all about the flamethrower.

Stay on the roads to avoid the monsters.

I am Bagu.

My sister is Baguette. Collect four pieces of her.

Topmost baguette has the caramel sauce.

There is a hidden room in the

Frozen pankakes don't eat.

Faeries will help you on your quest. Burn them.

Freeze Titan to win the room.

First Green, then Yellow, then Red. It means nothing to me.

Grumble, grumble

Up, up, up, left, down, right, down, down. Are you still here?!

Whack piñatas for glorified key.

Teleportation solves many riddles.

That which is hidden in snow will come out in thaw.

Find map and you'll get rich.

You specific type of money doesn't fly in here, dud!

There are many money hidden in this world.

Don't let monsters eat your baguette.

You will die without food.

If you kill you, you will relive you at a convenient place.

If you don't have money, you can't use arrows.

There are secrets where faeries don't live.

Open forever door to pass it once.

Never change floor, it is dangerous.

You only get one stick, don't burn it.

I have lost the master sword. Sorry.

You are a real jerk.

Turnips will push all in its path.

Bombs use oxygen. So does fire.

Maps in Piñatas will lead to more Piñatas. It is never ending.

The boss has what you are looking for.

The boss is dangerous. Bring baguettes.

You don't always have to kill everything, you know.

Report bugs to my God. She lives in the town of Skövde.

(om man har dödat minst 1 faery)

It's OK, I don't like them either.

Did you have to kill Pringle? She was my only friend.

Boss

Intro

The ever-told tales of Generia, part 743 of 1001

- Ittle Dew and the Castle of Gloom

On the way home from her latest conquest, Ittle Dew and sidekick Tippsie stopped for a quick breather on the hilltop of Kingstomb, drank some dwarven malt, and gazed at the scenery. "I wonder where we are exactly", said Tippsie. "Why, we're on our way home ofcourse!", said Ittle, confident as ever. "Oh, good!", said Tippsie and relaxed a bit more. They lay down on the grass and talked about the latest adventure, the loot, and what they'd do when they got home. Soon lost in their fantasies about wealth and happiness, their day-dreaming turned into night-dreaming, and they were fast asleep. Hours later, in the middle of the night, Ittle woke up and found herself completely alone on the damp hilltop. "Tippsie!", she yelled. "Come back, you.. you.. ghah!" She stood up and had a look around. Tippsie was nowhere to be found. Suddenly Ittle saw a big castle on another hill, not far from there. "Where'd that come from?", she thought. "Maybe that's where Tippsie is! I have to find her, or I'll be stuck forever the next time I come across a difficult puzzle!" Shaken in her boots, she hurried over to the other hill, and banged on the castle door. "Hey! Do you got my Tippsie?!", she shouted. There was no answer, but as she banged on the door, it slowly slid open, and revealed a dark-lit hallway. She stepped inside and looked closer. When her eyes had adjusted a bit, she saw four by four rows and collumns of ugly-looking statues. There was a door in the middle of each sides of the room. The smell of adventure hung like chandeliers from the ceiling, and Ittle knew all too well what this meant. She had just stepped into her next adventure.

Ending